UNCLASSIFIED

AD NUMBER

AD890303

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; NOV 1971. Other requests shall be referred to Air Force Weapons Lab., Kirkland AFB. NM.

AUTHORITY

AFWL ltr 16 Mar 1972

THIS PAGE IS UNCLASSIFIED

AFWL-TR-71-137

AFWL-TR-71-137

AD890303

A CALCULATION OF THE BLAST WAVE FROM THE CONSTANT VELOCITY DETONATION OF AN EXPLOSIVE SHEET

Gerald G. Leigh
Major USAF

TECHNICAL REPORT NO. AFWL-TR-71-137

November 1971

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base
New Mexico

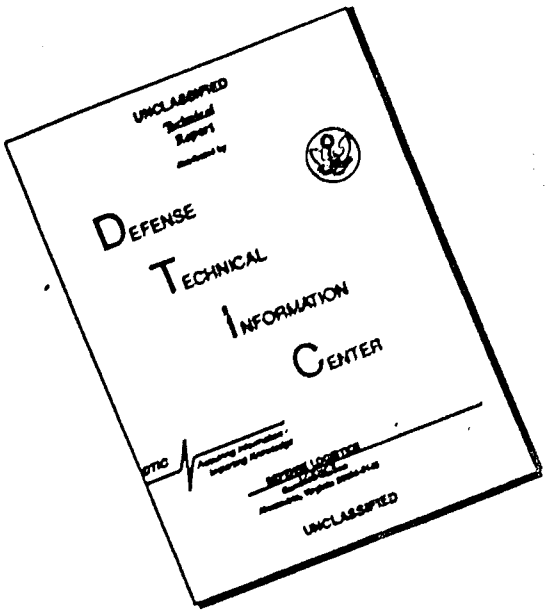
AD No.
DDC FILE COPY

DDC
RECEIVED
JAN 10 1972
B

Distribution limited to US Government agencies only because test and evaluation information is discussed in the report (Nov 71). Other requests for this document must be referred to AFWL (DEV), Kirtland AFB, NM, 87117.

274

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

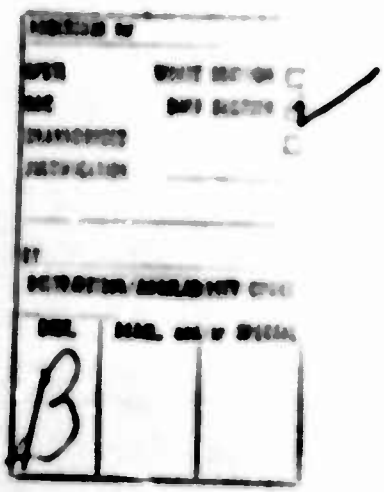
AFWL-TR-71-137

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base
New Mexico 87117

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report is made available for study with the understanding that proprietary interests in and relating thereto will not be impaired. In case of apparent conflict or any other questions between the Government's rights and those of others, notify the Judge Advocate, Air Force Systems Command, Andrews Air Force Base, Washington, DC 20331.

DO NOT RETURN THIS COPY. RETAIN OR DESTROY.



A CALCULATION OF THE BLAST WAVE FROM THE CONSTANT VELOCITY DETONATION OF AN EXPLOSIVE SHEET

Gerald G. Leigh
Major USAF

TECHNICAL REPORT NO. AFWL-TR-71-137

Distribution limited to US Government agencies only because test and evaluation information is discussed in the report (Nov 71). Other requests for this document must be referred to AFWL (DEV), Kirtland AFB, NM, 87117.

FOREWORD

This research was performed under Program Element 62601F, Project 06CB.

Inclusive dates of research were July 1969 through June 1971. The report was submitted 13 October 1971 by the Air Force Weapons Laboratory Project Officer, Major Gerald G. Leigh (DEV).

This report was presented to the Arizona State University Engineering Department in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

The author is deeply grateful to the members of his academic committee and particularly to his advisor, Dr. Donovan L. Evans, for the guidance, suggestions, and encouragement provided throughout this study. Acknowledgement is extended to Mr. Charles E. Needham of the Air Force Weapons Laboratory for his assistance in working with the SHELL and SHPLOT codes. Further acknowledgement is given Captain Gregg Canavan and Lieutenant Stephen Rockwood for their guidance and help in understanding the postdetonation combustion reactions. Acknowledgement is also due Arizona State University and the Air Force Institute of Technology for the opportunity to pursue these studies.

This technical report has been reviewed and is approved.

Gerald G. Leigh

GERALD G. LEIGH
Major, USAF
Chief, Facilities Survivability Branch
Project Officer

William B. Liddicoet

WILLIAM B. LIDDICOET
Colonel, USAF
Chief, Civil Engineering Research
Division

ABSTRACT

(Distribution Limitation Statement B)

Predictions of the blast wave from the constant velocity detonation of a thin explosive sheet have not previously considered the contribution of the post-detonation combustion of the detonation products. This problem is simplified by first considering the explosive material as releasing only energy into an ideal gas medium. This is solved for several energy densities and velocities of detonation. An expression that can predict the shape and location of the shock front from a line source of energy in a hypersonic flow is then developed. The problem is more realistically modelled by considering the explosive sheet to release six distinct mass species, as well as energy, into a mixture of nitrogen and oxygen. Results of this study shift the shock front to a position inside the previously determined shock front. Using the dissociation of molecular oxygen as the determinant for the reaction rate of combustion, and models for material mixing, the contribution of postdetonation combustion (PDC) to the total energy is determined. The mass and energy addition problems were then recomputed with PDC included. Results indicate that while significant amounts of energy are released by PDC, they are small compared to the energy released by the detonation, and there is essentially no contribution from PDC in the early time formation of the blast wave.

CONTENTS

ILLUSTRATIONS

ILLUSTRATIONS (cont'd)

ILLUSTRATIONS (cont'd)

ILLUSTRATIONS (cont'd)

TABLES

SECTION I

INTRODUCTION

The problem of predicting the blast effects from explosions has been of interest ever since the use of explosive materials began. Historically, the approach has been one of "trial and error." This has not been very successful until recent years when adequate electronic instrumentation became available. Calculational efforts were rare. Only in the last thirty years have calculations taken any degree of sophistication. Calculations of the blast effects from various explosive materials in several geometric arrays have been completed in the past ten years. Of particular recent interest are the blast effects from a large, relatively thin sheet of explosive material when the detonation is initiated uniformly along one edge and permitted to propagate along the length of the sheet. It is toward the theoretical determination of the blast effects from such an explosive arrangement that this work is directed.

BACKGROUND

Perhaps the most significant early calculation of airblast phenomena was accomplished by G. I. Taylor in 1941 (1). He considered an instantaneous release of a point source of energy in an ideal gas and was able to find a similarity solution by making a strong shock approximation. Sedov (2) and Von Neumann (3) later confirmed Taylor's results. Lin (4) and Sakurai (5) applied Taylor's technique to the line and

sheet source of energy. The development of an artificial viscosity concept by Richtmyer and Von Neumann (6) provided a means for numerically integrating across shock discontinuities, opening the way for use of modern numerical computation techniques in blast wave calculations. Brode (7) used this technique in his numerical treatment of the point source in ideal gas and was able to extend the solution to much lower overpressures where the strong shock approximation could not be applied. The same problem was addressed independently by Goldstine and Von Neumann (8) using a different numerical computation scheme. A repeat of the point source problem by Brode using a real equation of state for air (9) provided the first realistic calculation of a blast wave in air. Other calculations by Brode, such as the detonation of a sphere of TNT in air (10) and the expansion of a high-temperature, high-pressure sphere of hot gases in air (11), soon followed. All work described thus far has been one-dimensional, generally with spherical symmetry, and has led to the development of numerous one-dimensional hydrodynamic computer codes (12, 13, 14).

In the late 1950's, considerable interest in two-dimensional calculations began to appear. A major contribution by Harlow (15) was his development of the particle-in-cell (PIC) computing method wherein distinct particles of mass are permitted to move through an Eulerian grid in a manner satisfying the conservation equations. From this work have evolved several families of sophisticated two-dimensional

hydrodynamic computer codes useful in solving a variety of problems including hypervelocity impact, supersonic flow, blast wave propagation, etc. The SHELL family of hydrodynamic codes (16, 17) is representative. In the SHELL codes the basic computational scheme is similar to the PIC codes except that the mass distributed throughout the Eulerian grid is considered to be continuous with mass transport from cell to cell being accomplished as a continuum rather than as discrete particles. Numerous blast-wave calculations have been performed using the SHELL codes (18, 19, 20), and development to encompass a greater variety of problems continues.

Perhaps the most sophisticated family of two-dimensional computer codes is AFTON, developed by Trulio (21). These codes can be used for solving problems in continuum mechanics including the blast load deformation and motions of elastoplastic materials such as soil.

STATEMENT OF THE PROBLEM

In recent years large quantities of explosive materials have been used in the simulation of nuclear blast effects (22, 23). One simulation method of considerable interest utilizes detonating cord wrapped around a flat wooden rack and detonated in an earth cavity (24). It has been proposed to replace the detonating cord with large, relatively thin sheets of explosive materials. The blast-wave properties of sheet explosives are thus of immediate interest.

Consider a sheet of explosive material a few centimeters thick having an infinite length and width. A detonation is

initiated uniformly along one edge and allowed to proceed evenly along the sheet at the inherent velocity of detonation of the explosive material (Figure 1). Viewed along one edge, the sheet approaches a line, and the detonation becomes a point moving along this line. The problem is thus two-dimensional with a plane of symmetry along the center line of the explosive sheet (Figure 2).

The detonation proceeding along the explosive sheet is considered to be a classical Chapman-Jouguet detonation process (37). After proceeding some distance from the point of initiation, the process becomes essentially steady state. As a result of the detonation process, detonation products are produced and a large quantity of explosive energy is released. These hot, primarily gaseous, detonation products expand and compress the surrounding medium, which in most cases is air, resulting in the formation of a shock wave and a particle flow behind the shock. The shock waves moves with the point of detonation and when steady state is reached the shock wave assumes a fixed shape such that the properties behind the wave do not vary with time.

At the onset of expansion of the detonation products a distinct interface exists between these products and the surrounding air. However, as expansion proceeds, this interface becomes less distinct as mixing and diffusion take place. Finally, at considerable distance from the point of detonation, wholesale mixing has occurred and an essentially uniform mixture of gases exists over a large volume.

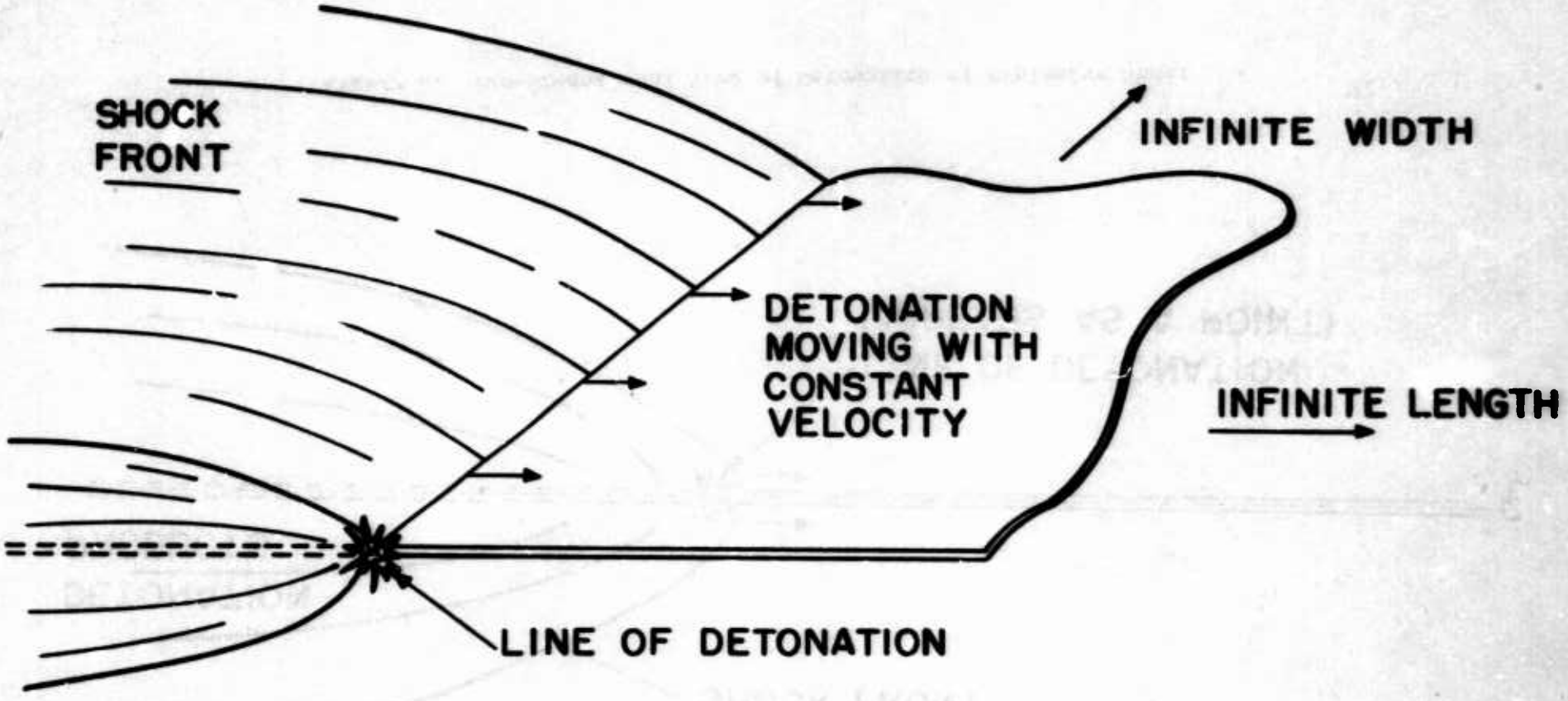


Figure 1. Detonation of Explosive Sheet

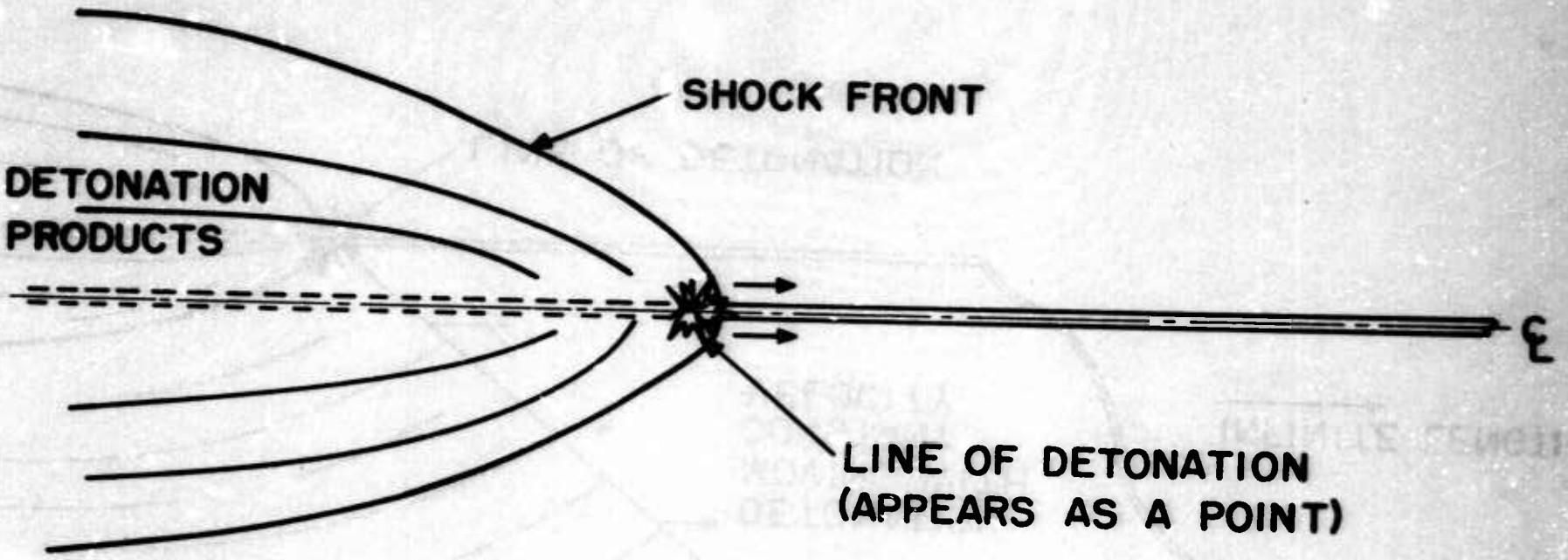


Figure 2. Two-Dimensional View of Detonation of Explosive Sheet

Most explosive materials (TNT, PETN, RDX, etc.) are highly oxygen negative, i.e., all available oxygen is consumed in the detonation reaction and excess combustible materials are released as detonation products. As these hot detonation products mix with the surrounding air, a post detonation combustion (PDC) process occurs and additional energy is released. However, the blast contribution from this PDC energy release is low because the release is distributed over a large spatial volume and over a long period of time. The extent to which this PDC affects the blast wave and the properties behind the shock front is not well known and of considerable interest.

APPROACH

Determining the blast wave properties associated with this explosive arrangement is first approached by idealizing the problem. The sheet of explosive material is considered to have no mass such that the line of detonation becomes a a constant line source of energy. The surrounding medium is considered to be an ideal gas. Consideration is given to a closed form analytical solution but little hope of finding such a solution exists. The idealized problem is then solved by using a finite difference computational technique (a SHELL hydrodynamic code). An empirical fit to the numerically obtained solution is obtained.

Next, a more realistic approach to the problem is taken. The explosive sheet is considered to be insignificantly thin, but to have finite mass and energy. A multimaterial

computational code is developed using a multimaterial equation of state. The properties and detonation products from a TNT detonation are used with this code to obtain a solution which accounts for the release of mass from the detonation process.

Finally, the post-detonation combustion of the detonation products by the oxygen in the surrounding air is considered. A simplified combustion scheme is incorporated into the multimaterial code above, and the previous problems are solved with PDC included.

The results of the above three computational schemes are compared and the effect of PDC on the shape and location of the blast wave is determined.

SECTION II

THE IDEALIZED PROBLEM

To examine the blast wave from a sheet explosive in its simplest form we make several idealizing assumptions. Looking at the sheet along one edge (2 dimensions), we first consider the explosive to have no mass and to be a pure source of energy concentrated along a line. The line of detonation becomes a line source of energy perpendicular to the x-y plane (see Figure 3) and appears as a point moving along the plane of symmetry at a fixed velocity and releasing a constant amount of energy for each unit of length that it traverses. The medium through which this point source moves is considered to be an ideal gas. Further clarification can be gained by changing coordinates, fixing the line of detonation in space, and letting the background flow past this point. Since most velocities of detonation (10,000 to 20,000 fps) well exceed the sonic velocity in the surrounding medium the flow produced by the change of coordinates is hypersonic ($M > 10$). The problem is thus reduced to the two-dimensional plane symmetric case of a constant line source of energy in a steady hypersonic flow.

ANALYTIC SOLUTION

The surrounding medium is considered to be an inviscid, compressible ideal gas having a specific heat ratio of 1.4. Heat conduction and gravitational effects are ignored. The governing equations for steady flow are

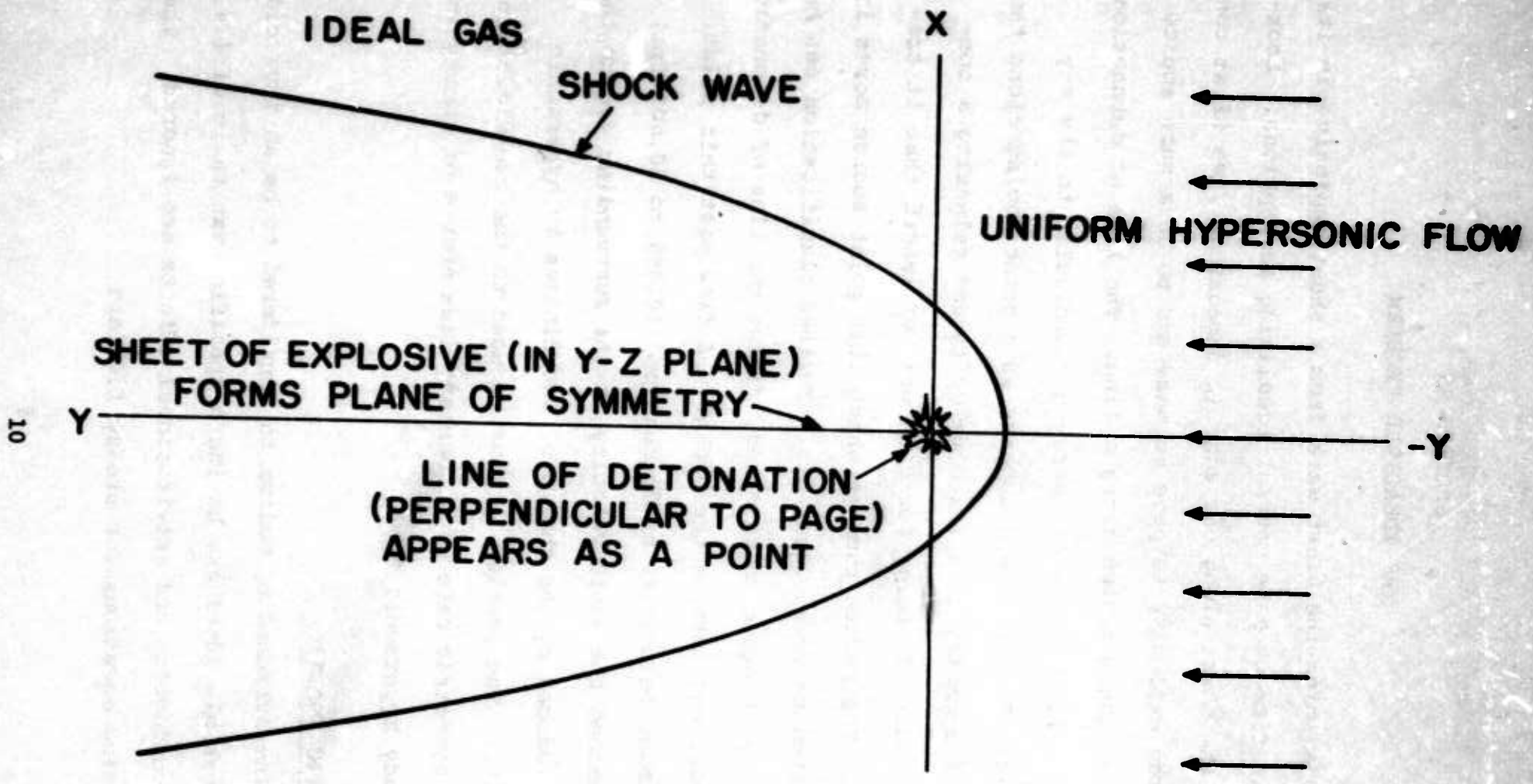


Figure 3. Line Source of Energy in Hypersonic Flow

Continuity

$$\frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0 \tag{1}$$

Momentum

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + \frac{1}{\rho} \frac{\partial P}{\partial x} = 0 \tag{2}$$

$$u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + \frac{1}{\rho} \frac{\partial P}{\partial y} = 0 \tag{3}$$

Energy

$$\rho u \frac{\partial e}{\partial x} + \rho v \frac{\partial e}{\partial y} + P \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = E \tag{4}$$

State

$$e = \frac{P}{(\gamma - 1)\rho} \tag{5}$$

where

E is the energy deposited per unit volume per unit time

u and v are the velocities in the x and y directions

P is the pressure

ρ is the density

e is the specific internal energy.

These equations can be combined and reduced to the following form:

$$u^2 - \frac{\gamma P}{\rho}\frac{\partial u}{\partial x} + uv \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + v^2 - \frac{\gamma P}{\rho}\frac{\partial v}{\partial y} = (1 - \gamma)\frac{E}{\rho} \tag{6}$$

This is a nonlinear, inhomogeneous, 1st order differential equation with four dependent variables (u, v, P, ρ) and two independent variables (x, y).

Numerous approaches to finding an exact or a nonnumerical approximate solution to these equations have been attempted, but so far without success. The methods tried are described below.

Similarity Approach -- The 1-D, unsteady flow resulting from the instantaneous release of a fixed quantity of energy at a point, along a line, or on a plane is self-similar during the early times when the blast pressures are large compared to the ambient pressure. Numerous investigators (1, 2, 3, 4, 5) have utilized this characteristic to reduce the number of independent variables and thus obtain solutions to blast-wave problems. A similarity solution to the present problem was considered. Following the method of Sedov as described by Korobeinikov (26), the dimensionality of all the variables and parameters in this problem was considered. The number of constants with independent dimensions which could be formed was found to be three. The problem was not self-simulating, then, according to the criteria defined by Korobeinikov, and further consideration of a similarity solution was abandoned.

Hypersonic Flow - Point Explosion Approach -- During the literature search several papers describing the use of point source explosions to simulate hypersonic flow past various bodies were encountered (27, 28). Consideration was given to

using one of these methods with the present problem. Hypersonic flow simulation is based on an "equivalence principle" stated by Hayes and Probstein (29, pg. 37) as the "the equivalence of a 3-D physical problem to a 2-D unsteady problem with a parameter or a 2-D or axisymmetric physical problem to a 1-D unsteady problem with a parameter." Stated in another way, the "equivalence principle" is

> "The flow as viewed in any transverse plane is independent of the flow in any other transverse plane. In the 2-D physical case we may liken the flow in a given transverse plane to the flow in a cylinder driven by a piston."

Thus, in the 2-D hypersonic flow over a body, the flow in any transverse plane can be considered as a function of one space dimension and time; the flow in this plane is self-similar. Therefore, it is the same as the flow outward from a point or line source of energy and different energy sources can be used to simulate different bodies in hypersonic flow.

When these concepts were brought to bear on the present problem, it was apparent that the flow in any transverse plane in the present problem is not self-similar and is not independent of any other transverse plane, at least not near the origin or the region between the origin and the forward-most portion of the shock wave.

Hodograph Transformation Approach -- A method in common use for solving compressible flow problems is that of the hodograph transformation as explained by Von Mises (30). In this method, the dependent and the independent variables are interchanged by using a Jacobian of the form

$$J = \frac{\partial x}{\partial u}\frac{\partial y}{\partial v} - \frac{\partial x}{\partial v}\frac{\partial y}{\partial u} \tag{7}$$

When this Jacobian is applied to a homogeneous equation similar to the equation

$$\left(u^2 - \frac{\gamma P}{\rho}\right)\frac{\partial u}{\partial x} + uv\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) + \left(v^2 - \frac{\gamma P}{\rho}\right)\frac{\partial v}{\partial y} = 0 \tag{8}$$

one obtains

$$\left(u^2 - \frac{\gamma P}{\rho}\right)\frac{1}{J}\frac{\partial y}{\partial v} - \frac{1}{J}uv\left(\frac{\partial x}{\partial v} + \frac{\partial y}{\partial u}\right) + \left(v^2 - \frac{\gamma P}{\rho}\right)\frac{\partial x}{\partial u}\frac{1}{J} = 0 \tag{9}$$

and since the equation is homogeneous one can multiply out the Jacobian to obtain a linear equation

$$\left(u^2 - \frac{\gamma P}{\rho}\right)\frac{\partial y}{\partial v} - uv\left(\frac{\partial x}{\partial v} + \frac{\partial y}{\partial u}\right) + \left(v^2 - \frac{\gamma P}{\rho}\right)\frac{\partial x}{\partial u} = 0 \tag{10}$$

When this approach is tried with the present problem the Jacobian cannot be multiplied out since equation 6 is not homogeneous. One obtains

$$\left(u^2 - \frac{\gamma P}{\rho}\right)\frac{\partial y}{\partial v} - uv\left(\frac{\partial x}{\partial v} + \frac{\partial y}{\partial u}\right) + \left(v^2 - \frac{\gamma P}{\rho}\right)\frac{\partial x}{\partial u} = J(1 - \gamma)\frac{E}{\rho} \tag{11}$$

where $J = J(x, y)$ and is unknown.

We consider the possibility of evaluating the term $J(1 - \gamma)E$ realizing that E is very small at all locations except near the origin.

$$J = \frac{1}{J} = \frac{1}{\frac{\partial u}{\partial x}\frac{\partial v}{\partial y} - \frac{\partial u}{\partial y}\frac{\partial v}{\partial x}} \tag{12}$$

Consideration of the physics of the problem leads to the conclusion that near the origin

$$\frac{\partial v}{\partial x} = 0 \qquad \frac{\partial v}{\partial y} = \text{some finite number}$$

$$\frac{\partial u}{\partial x} = 0 \qquad \frac{\partial u}{\partial y} = \text{some finite number} \tag{13}$$

which results in $j = 0$ and the term $J(1 - \gamma)\frac{E}{\rho}$ goes from essentially zero at all points not near the origin to an infinite value near the origin. The reasoning behind the conclusion that $\partial u/\partial x = 0$ near the origin is that along the x-axis the x-component of velocity drops from the hypersonic value before the shock to a subsonic value immediately behind the shock. Then, since it is flowing against an increasing pressure gradient as it approaches the point source of energy, it continues to decrease in magnitude until it reaches the origin where the line source of energy is located. Then beyond the origin, the x-component of velocity begins to increase because the pressure is now decreasing with increasing x. Thus, $\partial u/\partial x$ must be zero near the origin.

From the above analysis it was concluded that the hodograph transformation could not be used to solve the presented problem.

Von Mises' Approach for a Nonisentropic Hypersonic Flow Around a Body -- Consideration was given to solving the problem of hypersonic flow about a point source using a nonisentropic method described by Von Mises (30). In this method, Von Mises uses the continuity equation and the equation for

the y-component of momentum. He introduces a stream function and then converts these equations to natural coordinates (s, n) that follow the streamlines. That is, the s coordinate is tangent to the streamline and the n coordinate is normal to the streamline at any point in the flow field. He is thus able to obtain an expression relating ψ, P, q, ρ_o, s and n where q is the component of velocity tangent to the streamline (magnitude of the velocity vector), ψ is the stream function, and ρ_o is a constant density at some point in the flow field. He then recognizes that while behind a curved shock the entropy is not constant, it is constant along any streamline (for flow around a body) and he uses this fact to obtain an expression relating P and n. After much algebra he is able to obtain an expression of the form

$$\left[1 - \frac{q_x^2}{A^2}\right]\frac{\partial^2\psi}{\partial x^2} - \frac{2q_x q_y}{A^2}\frac{\partial^2\psi}{\partial x\partial y} + \left[1 - \frac{q_y^2}{A^2}\right]\frac{\partial^2\psi}{\partial y^2}$$

$$= \frac{\rho^2}{\rho_o^2} G^1 - \frac{\rho F^1}{\rho_o^2}\left[\rho T + \frac{M^2}{\partial S/\partial P}\right] \qquad (14)$$

where

q_x, q_y are the respective components of velocity

A is the local speed of sound

M is the local Mach number

G^1 and F^1 are known functions depending on the boundary conditions at the shock.

For a known material (such as a perfect gas) and for flow around a defined body this equation can be solved to obtain the entire flow field.

In attempting to apply this approach to the present problem difficulty was immediately encountered with the entropy equation. Whereas entropy may be constant along a streamline for some of the flow field, there is at least one streamline (at least the one passing through the origin) where entropy is not constant, therefore, this concept could not be applied. It was then decided to try the energy equation (equation 4) to obtain relations between P and n in a manner similar to Von Mises'. However, the energy equation could not be converted to natural coordinates because of the inhomogeneous term containing E which is a function of the spatial coordinates and not of the velocity field.

Approximate Method of Integral Representations -- During the literature search, a Soviet paper by P. I. Chushkin and Li Li-kan (31) was discovered in which they used an approximate method of integral representation to determine the 2-D hypersonic flow of gas about a body. Their method consisted of consolidating the continuity equation, the y-component of momentum equation and the entropy equation ($Ds/Dt = 0$) into a single expression.

$$\frac{\partial P_i}{\partial x} + \frac{\partial Q_i}{\partial y} = T_i \qquad (i = 1, 2, 3) \qquad (15)$$

where

$$P_1 = y^\nu \rho u \qquad Q_1 = y^\nu \rho v \qquad T_1 = 0$$

$$P_2 = y^\nu \rho uv \qquad Q_2 = y^\nu (\rho v^2 + p) \qquad T_2 = \nu\rho$$

$$P_3 = y^\nu \rho uS \qquad Q_3 = y^\nu (\rho vS) \qquad T_3 = 0$$

and $S = p/\rho^2$ and $\nu = 0, 1$ for plane or axisymmetric flow, respectively.

$$\xi = \frac{y - y_o(x)}{y_N(x) - y_o(x)} = \frac{y - y_o(x)}{\varepsilon(x)}$$

where $y_o(x)$ is the y distance to the body surface at any x and $y_N(x)$ is the y distance to the shock wave at any x and $\varepsilon(x) = y_N(x) - y_o(x)$ is the distance between the body and the shock wave and is a function of x only. The result of this transform applied to equation 15 is

$$\frac{\partial}{\partial \xi}[Q_i - P_i(\Sigma\varepsilon' + y_o')] + \frac{\partial (P_i \varepsilon)}{\partial x} = T_i \varepsilon \qquad (i = 1, 2, 3) \quad (16)$$

They next divide the flow region between $y_o(x)$ and $y_N(x)$ into bands of equal width designated by ξ_o, ξ_1, ξ_2 $\cdots$ ξ_N where ξ_o is the body surface and $\xi_N = 1$ is the shock wave. They then integrate equation 16 over these bands to obtain

$$\frac{d}{dx} \varepsilon \int_o^{\xi_1} P_i d\xi - P_{i1}(\xi_1 \varepsilon' + y_o') + Q_{i1} + P_{io} y_o' - Q_{io} = \varepsilon \int_o^{\xi_1} T_i d\xi \qquad (17)$$

and

$$\frac{d}{dx}\,\varepsilon\int_{\xi_{n-2}}^{\xi_n} P_i\,d\xi - P_{in}(\varepsilon'\xi_n + y_o') + Q_{in} + P_{i,n-2}(\varepsilon'\xi_{n-2} + y_o')$$

$$- Q_{i,n-2} = \varepsilon\int_{\xi_{n-2}}^{\xi_n} T_i\,d\xi \qquad \begin{matrix}(i = 1,2,3)\\(n = 2,3,\cdots,N)\end{matrix} \qquad (18)$$

In the above they have used Leibnitz' rule to obtain

$$\int_{\xi_{n-2}}^{\xi_n} \frac{\partial}{\partial x}(P_i\varepsilon)\,d\xi = \frac{d}{dx}\,\varepsilon\int_{\xi_{n-2}}^{\xi_n} P\,d\xi - P_i(x,\xi_n)\frac{d\xi_n}{dx} + P_i(x,\xi_{n-2})\frac{d\xi_{n-2}}{dx} \qquad (19)$$

and have assumed the last two terms to be approximately equal for n-2 close enough to n.

They then approximate the integrands in the above expressions with 2nd-order polynomials and perform the designated differentiation with respect to x. They thus arrive at a series of ordinary differential equations which can be solved for the slopes of the various flow variables.

In attempting to apply this method to the present problem it was first discovered that the governing equations could not be consolidated into the same general form as equation 15 because $Ds/Dt \neq 0$. The energy equation introduces additional terms in equation 15 (for $i = 3$). Nevertheless, an expression similar to equation 15 is obtained, differing only for $i = 3$.

The transformation, $\xi = (y - y_o)/\varepsilon$ was applied and an equation similar to equation 16 obtained. For $i = 3$ it has the form

$$\frac{\partial}{\partial\xi}[Q_3 - P_3(\xi\varepsilon')] + (\gamma - 2)\frac{\rho}{\rho}\frac{\partial}{\partial\xi}[(u\xi\varepsilon') - v] + \frac{\partial}{\partial x}(P_3\varepsilon)$$

$$+ (\gamma - 2)\frac{\rho}{\rho}\frac{\partial}{\partial x}(u\varepsilon) - 2(\gamma - 2)\frac{\rho}{\rho}u\varepsilon' = \varepsilon(\gamma - 1)E \qquad (20)$$

while for $i = 1, 2$, it was identical with equation 16.

In studying the approximation made by dropping the last two terms of equation 19, it can be seen that this approximation is only valid at some distance away from the head of the body. In the region immediately behind the curved shock this approximation is not valid because $d\xi_n/dx$ and $d\xi_{n-2}/dx$ are both large and different. Thus, in the present problem where we are interested in determining the flow immediately behind the curved portion of the shock, we cannot use the approximation. Chushkin and Li-kan avoid this problem by picking up the solution some distance downstream after some other method has been used to obtain the flow around the nose of the body. Even then, an attempt was made to integrate equation 20 in a manner similar to Chushkin and Li-kan. In so doing, one obtains a series of integrals of the form

$$\int_{\xi_{n-2}}^{\xi_n} \frac{\rho}{\rho}\frac{\partial}{\partial\xi}[(u\xi\varepsilon') - v]d\xi$$

$$\int_{\xi_{n-2}}^{\xi_n} \frac{p}{\rho} \frac{\partial}{\partial x}[u\varepsilon]d\xi \qquad (21)$$

$$\int_{\xi_{n-2}}^{\xi_n} \frac{p}{\rho} u\varepsilon' d\xi$$

No method of solving these integrals was discovered and methods of approximating them soon became too cumbersome to be practical.

Termination of Search for Exact Solution -- In view of the fact that this idealized problem was only an introductory portion of the planned dissertation study, it was decided that numerical methods and computerized solutions were the only practical way to proceed.

NUMERICAL SOLUTION

The idealized problem (line source of energy in a hypersonic flow) was next addressed numerically using a hydrodynamic computer code named SHELLTC. In this code the 2-D hydrodynamic equations of continuity, momentum and energy for an inviscid compressible fluid plus an ideal gas equation of state are converted to finite difference form and utilized in a computation algorithm which, in essence, integrates these equations in a time-marching sequence. Pressure, internal energy, mass and two components of velocity are the variables accounted for throughout the flow. (See Appendix A for detailed discussion of SHELLTC.)

The computational solution is started by establishing a grid in cartesian coordinates into which an initial quantity of mass is placed. The mass is distributed uniformly throughout all the cells of the grid and an initial velocity, density and internal energy is assigned to the mass in each cell. This initial loading of the grid represents the background hypersonic flow of the line source problem where the plane of symmetry of the grid represents the center of the infinitely thin sheet of explosive energy (Figure 4). An auxiliary generator code titled CLAMTC is used to construct the grid and to establish the initial flow conditions. The code records this information as the zero cycle on a magnetic tape (the problem tape) which is then used by SHELLTC throughout the computation.

In the computational scheme of SHELLTC, time moves forward in small increments and a small amount of mass transport is accomplished with each time step. Thus, in the background flow, mass moves uniformly from right to left, with the incoming flow from the right being reestablished in the first two rows of cells at the beginning of each time step. Mass is permitted to flow out the left and top boundaries of the grid as dictated by the flow computations.

After the satisfactory background flow is established, deposition of energy begins at the point of detonation. The amount of energy deposited is determined by the strength of the energy source (energy content of the explosive sheet), the detonation velocity and the length of the time step.

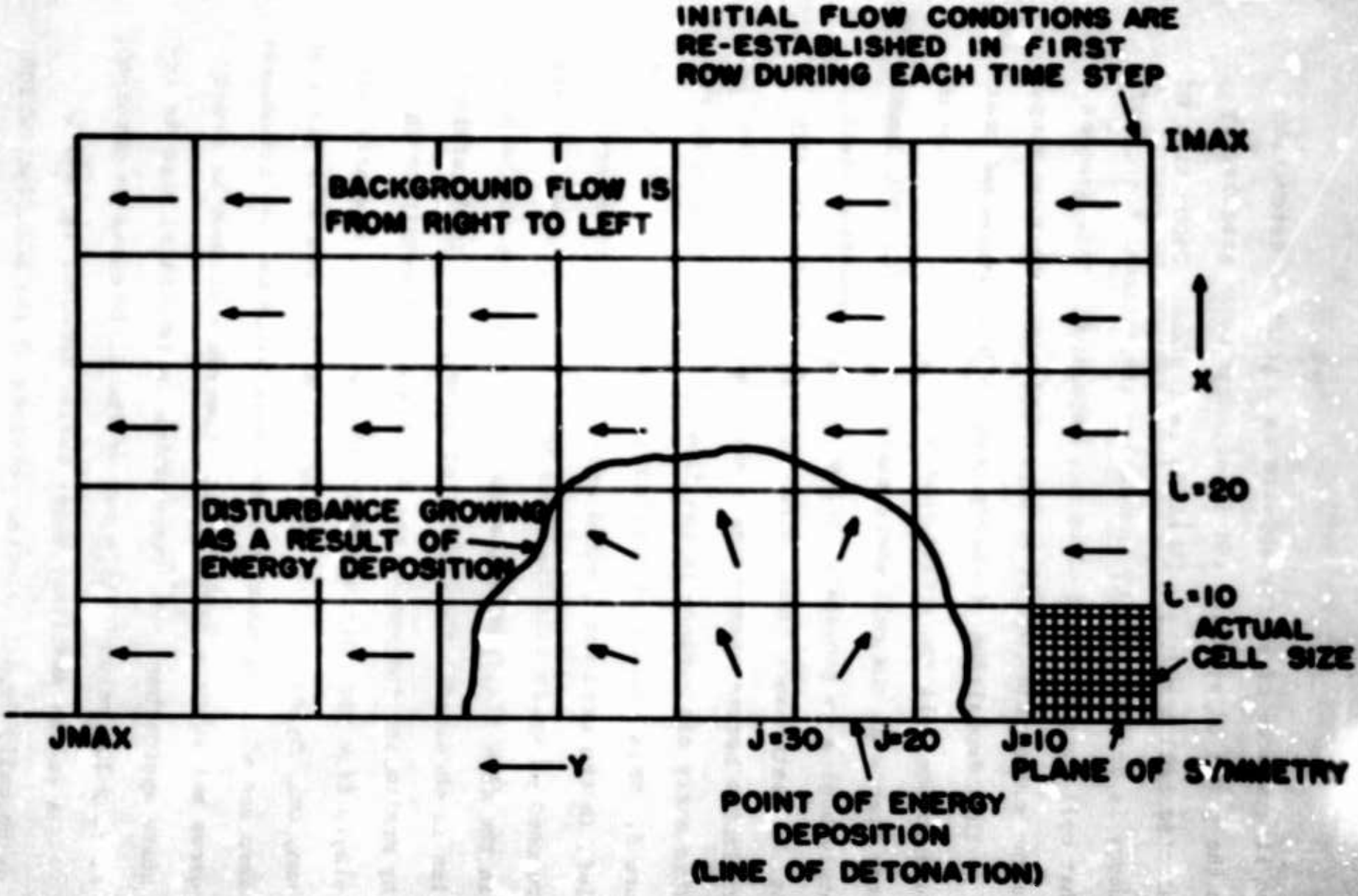


Figure 4. Computational Grid

The line source of energy appears as a point of detonation on the plane of symmetry and is adjacent to an arbitrarily selected cell (i = 1, j = 15) along that plane. Deposition of energy is accomplished by increasing the internal energy of that cell by the amount of energy deposited. This process, in the subroutine CDT, is accomplished once each time step.

This deposition of energy results in an increased pressure in the cell and a disturbance in the flow. The disturbance starts at the cell where energy is deposited but immediately begins to propagate farther out into the flow field. As the disturbance grows, dramatic variations in the flow variables become apparent as a shock wave begins to form. This early disturbance in the flow is clearly visible in Figure 5. This shock wave continues to move outward and to the left in the grid until steady-state flow conditions are reached at which time the shock wave becomes a standing wave in the flow field and the steady-state solution to the problem is obtained. The arrival of steady state is determined by monitoring the sum of the differences in magnitude in a single flow variable (pressure was used) along one grid column, cell by cell. This sum of the differences will start at zero and will increase to some large value as the disturbance grows but returns exponentially towards a minimum as steady state approaches. The computation is terminated when an arbitrary low value of this sum of the differences is obtained.

A second auxiliary code, called SHPLOTL, is used to graphically depict selected portions of the solution, which

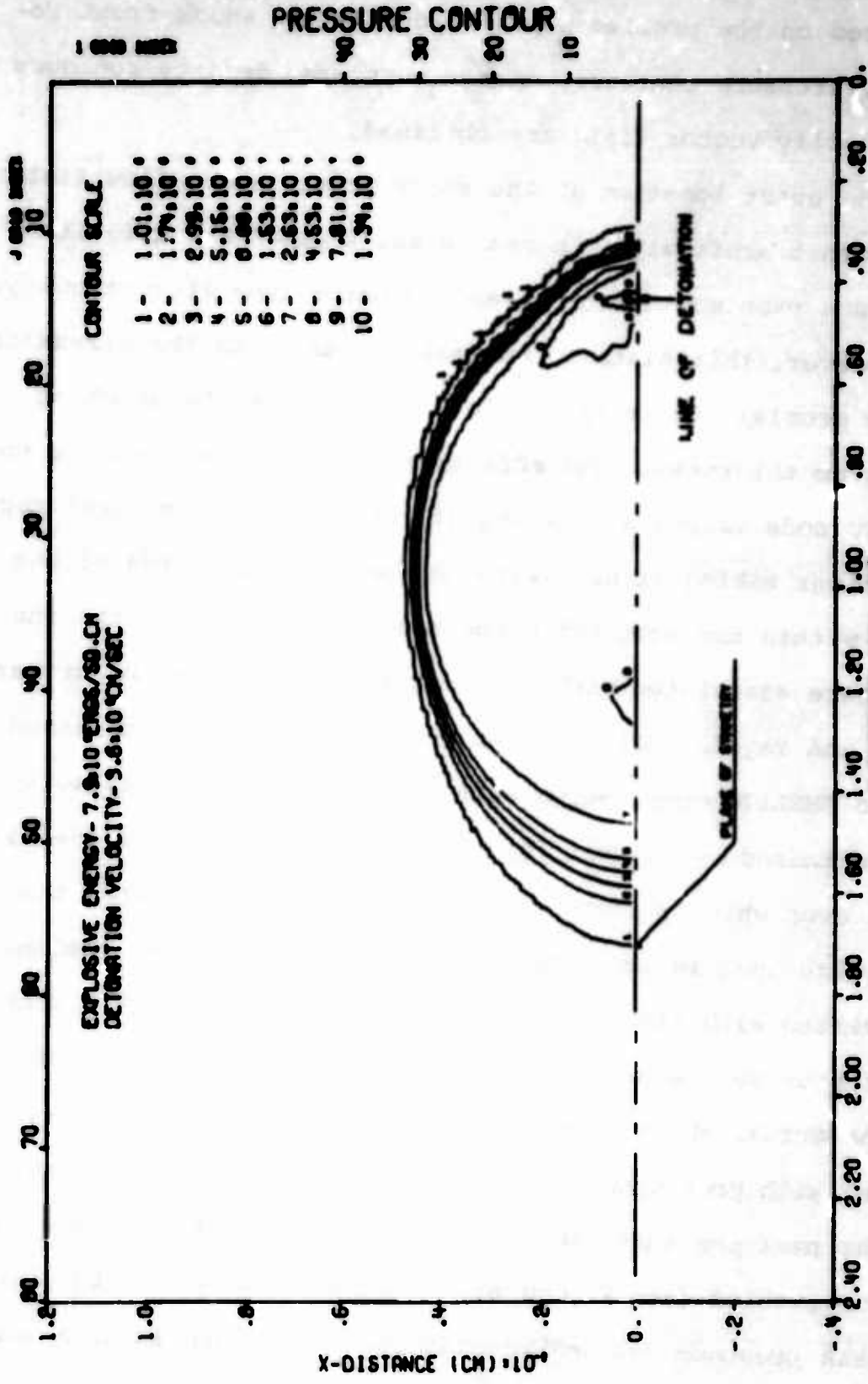


Figure 5. Pressure Contour

SHELLTC CALCULATION OF POINT SOURCE IN HYPERSONIC FLOW
TIME .000160 SEC CYCLE 2185 PROBLEM 21.000

is stored on the problem tape. Plots of the shock front location, pressure contours, energy contours, density contours and velocity vector field are obtained.

The exact location of the shock front in the flow field is somewhat arbitrary. In real gases a shock has a finite structure over some distance and is not a true discontinuity. If, however, this distance is small relative to the dimensions of the problem, it is appropriate to consider the shock as having no thickness. The effective viscosity inherent in the SHELLTC code causes a smearing of the shock over several zone dimensions making it necessary to specify a position of the shock within the computed structure. In a study of the shock structure associated with a one-dimensional planar blast wave, Leigh and Yagala (32) compared the shock structure obtained from a SHELLTC computation with the planar similarity solution obtained by Sakurai (5). This work showed that the distance over which a shock was spread was dependent upon the zone size used in the computation and the effective viscosity associated with the code. The location of the shock discontinuity as determined from the similarity solution was chosen as the actual shock location and occurred at positions associated with pressures of between 60 percent and 80 percent of the peak pressure for cell dimensions suitable to the line source problem (see Figure 6). A shock location at 80 percent of peak pressure was arbitrarily chosen for use in this work. The plotting code (SHPLOTL) was adapted to search for the 80 percent of peak pressure point normal to the pressure gradient

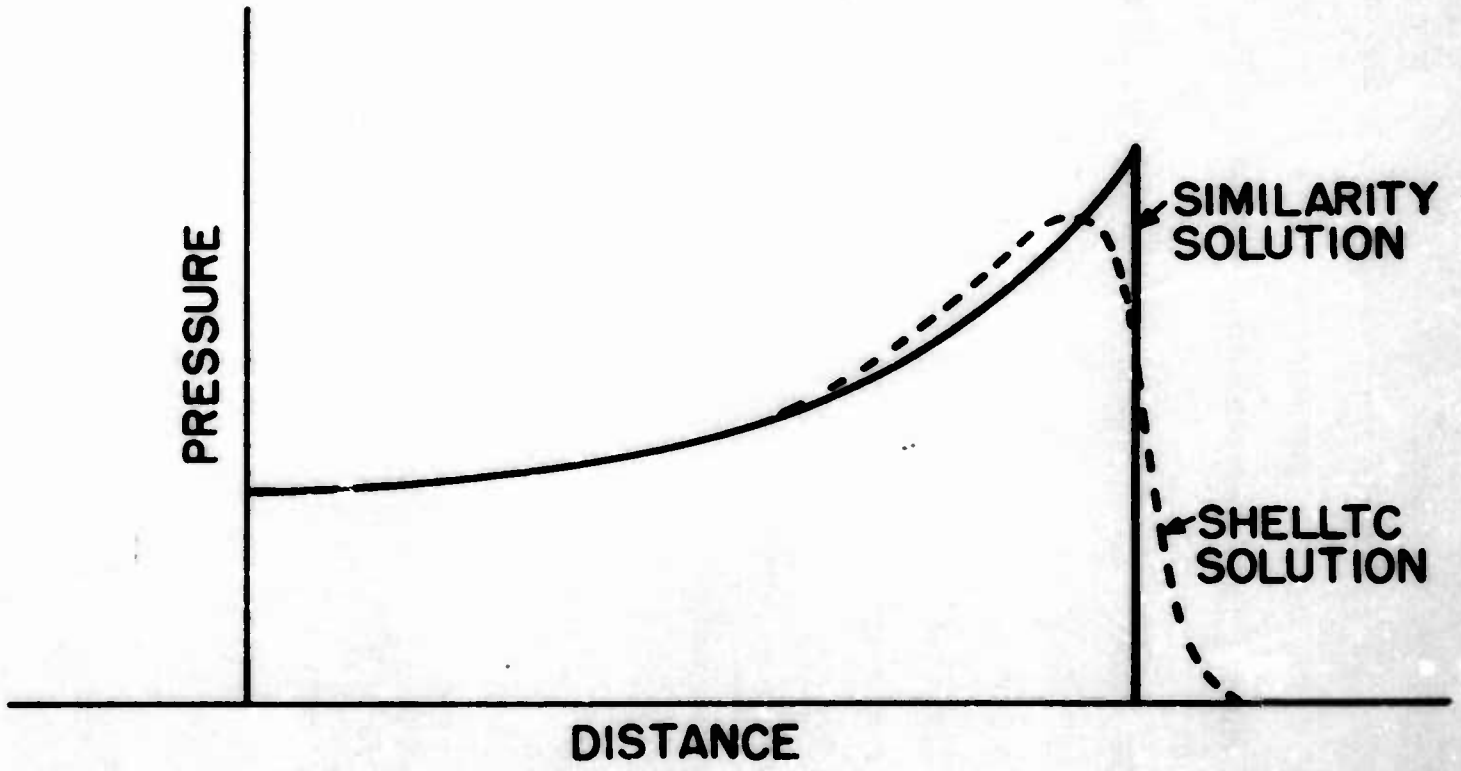


Figure 6. Comparison of One-Dimensional Blast Wave Solutions

and form all such points into a smooth curve. This curve is considered to be the location of the shock front.

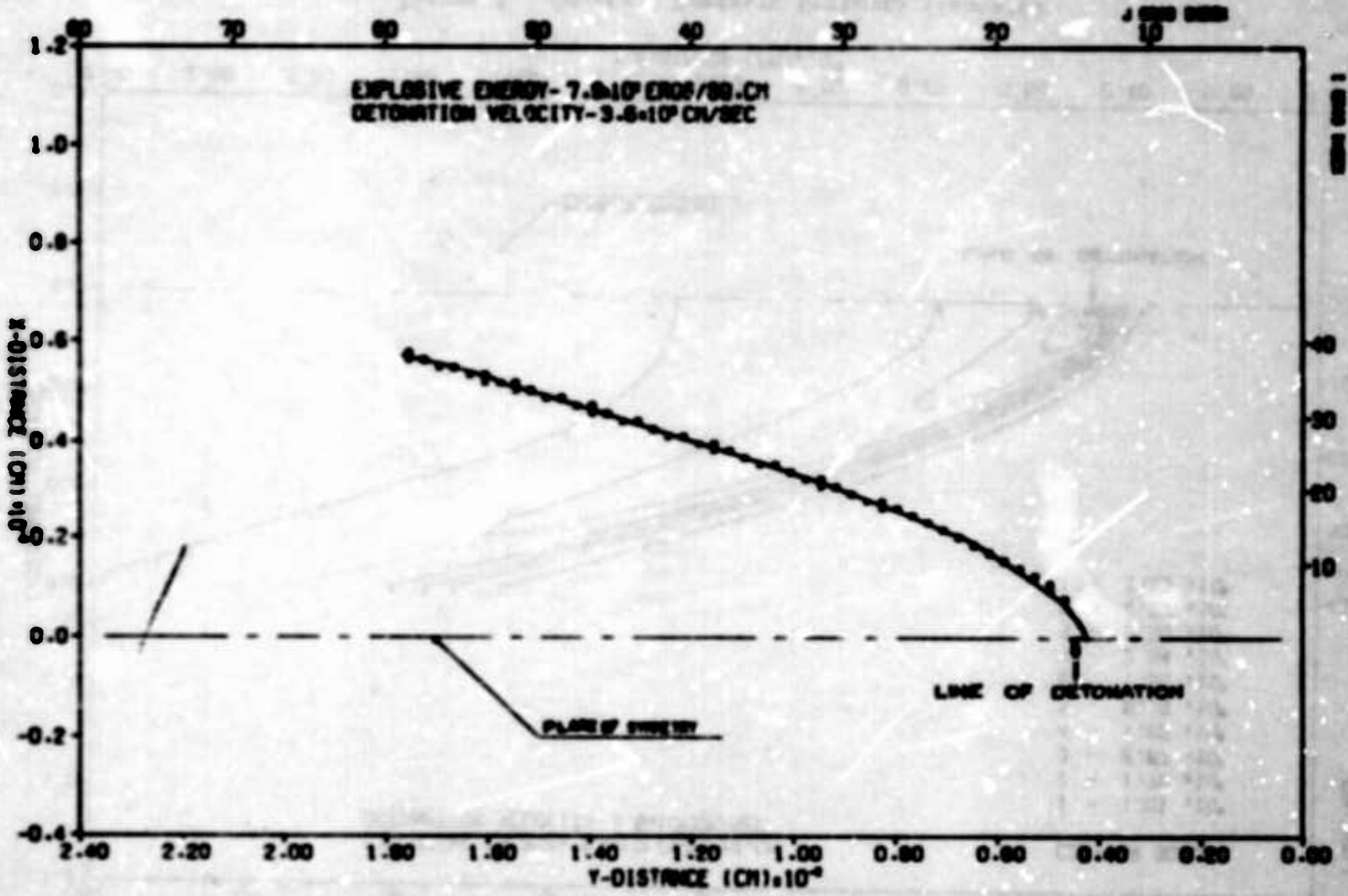
RESULTS OF NUMERICAL SOLUTION

In order to solve the point source in hypersonic flow problem, four values of source energy and detonation velocities were chosen for computational solutions.

Energy (ergs/cm^2)	Detonation Velocity (cm/sec)
7.9×10^6	7.2×10^5
7.9×10^7	3.6×10^5
7.9×10^8	2.4×10^5
7.9×10^9	1.8×10^5

The selection of velocity was based on an assumed velocity of available detonating sheet explosives (7.2×10^5 cm/sec) and fractions thereof. The explosive energy values were obtained by considering sheets of explosive material 1/4-inch thick having energy densities of 10^7 through 10^{10} ergs per gram and a density of 1.25 gram per cubic centimeter. These choices were quite arbitrary but were influenced by the required computation time, the greater the energy deposition the larger the computation time required. The computer runs were made using a CDC-3400 for the low energy levels and a CDC-6600 for the higher energy levels.

The results for the energy value of 7.9×10^8 ergs/cm^2 and velocity of 3.6×10^5 cm/sec are shown in Figures 7 through 12. A 240-cm long by 60-cm high grid having cell sizes 3 cm long by 1.5 cm high was used in this computation.



SHELLTC CALCULATION OF POINT SOURCE IN HYPERBOLIC FLOW
TIME 0.0000002 SEC CYCLE 800 PROBLEM 21.000

Figure 7. Shock Front Location in the XY Plane

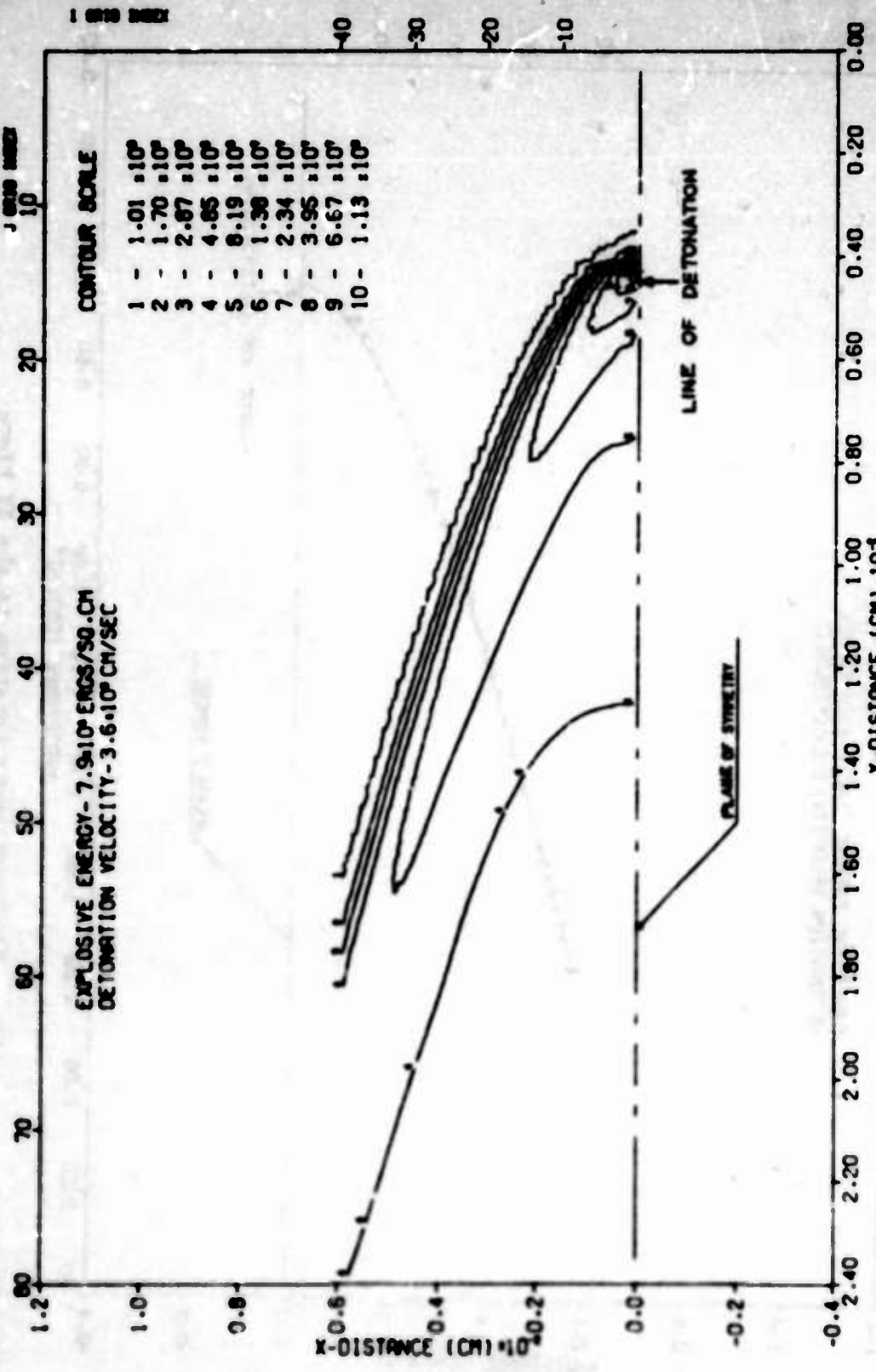


Figure 8. Constant Pressure Contours (Isobars)

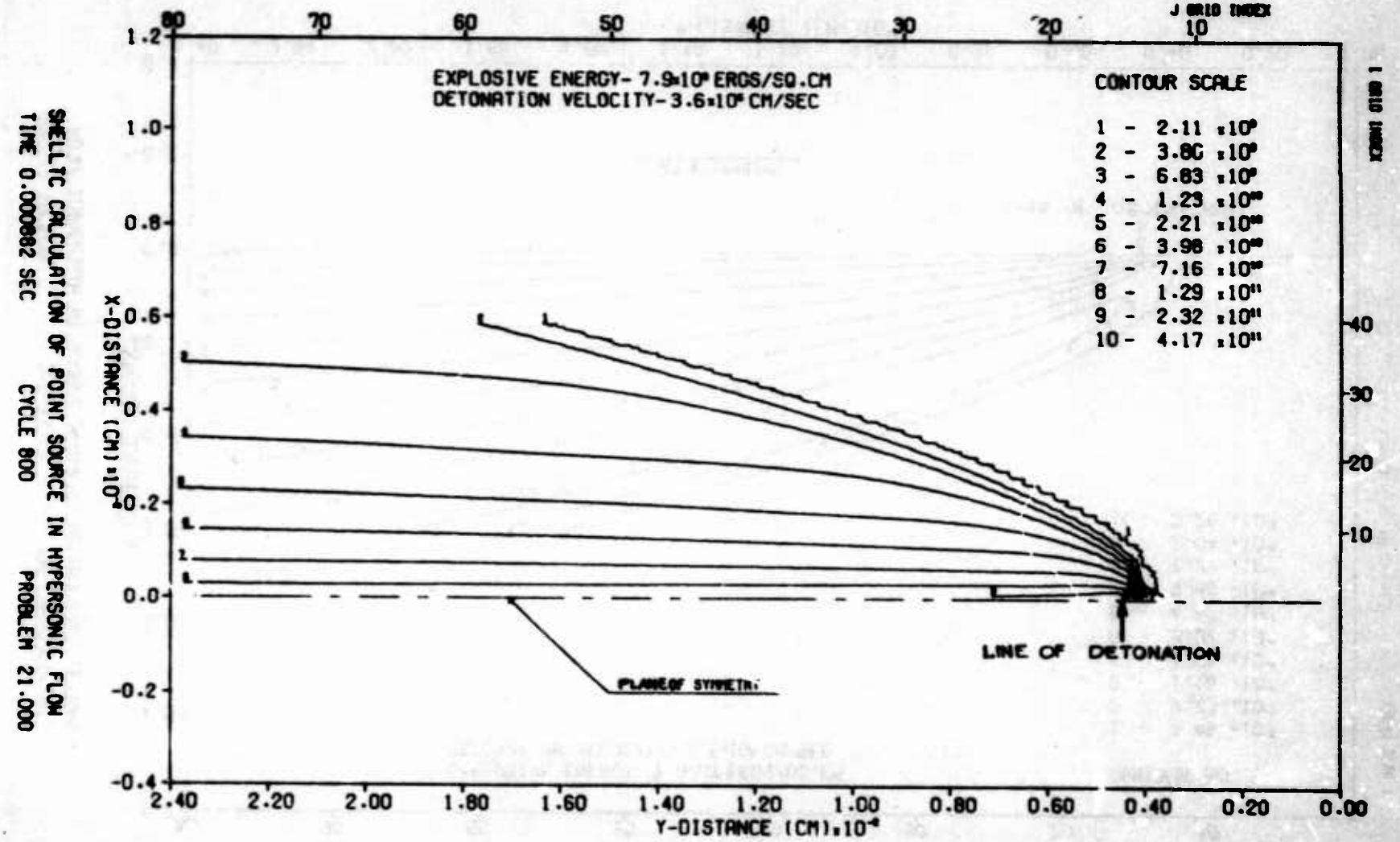


Figure 9. Constant Energy Contours

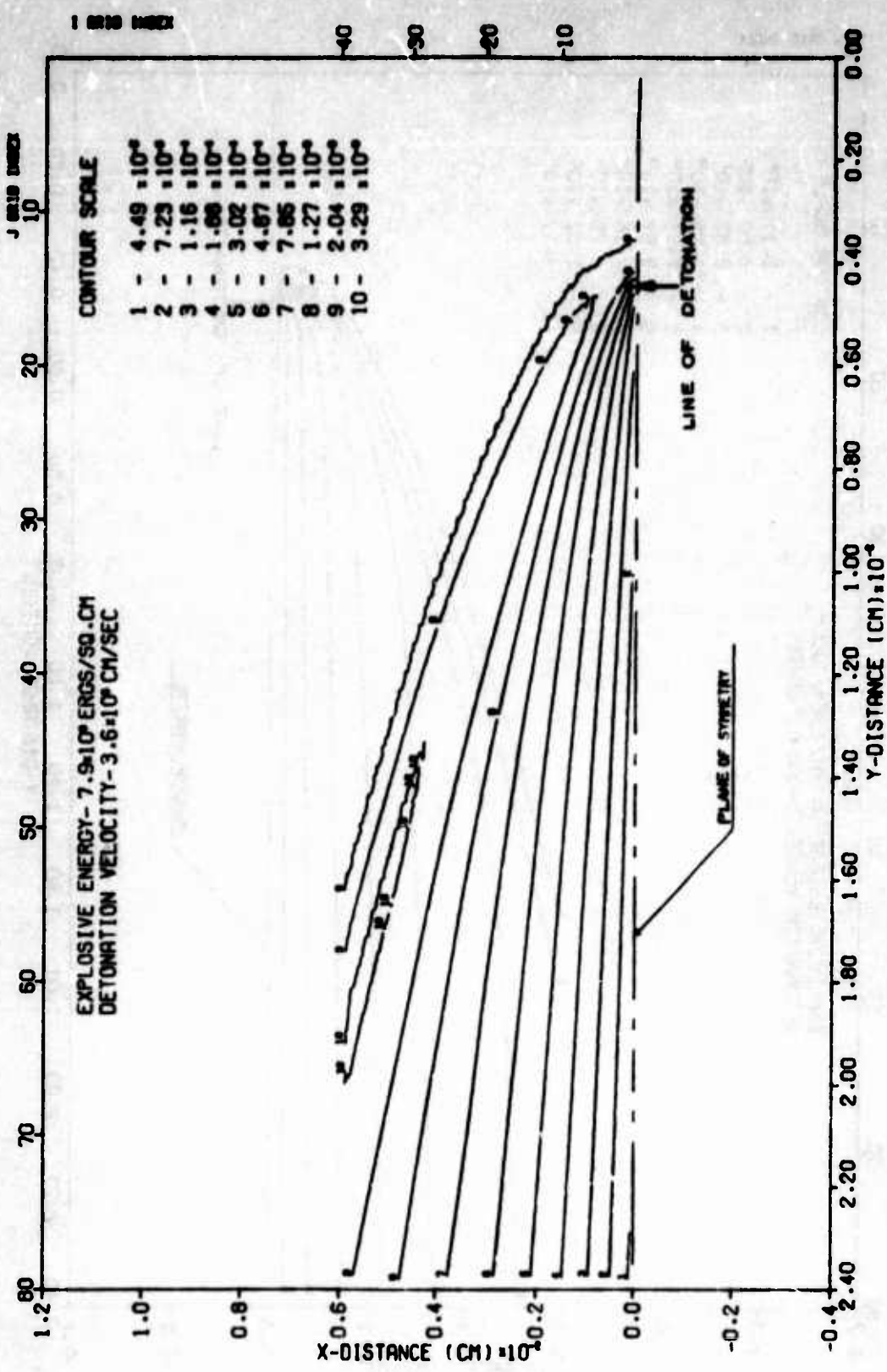


Figure 10. Constant Density Contours

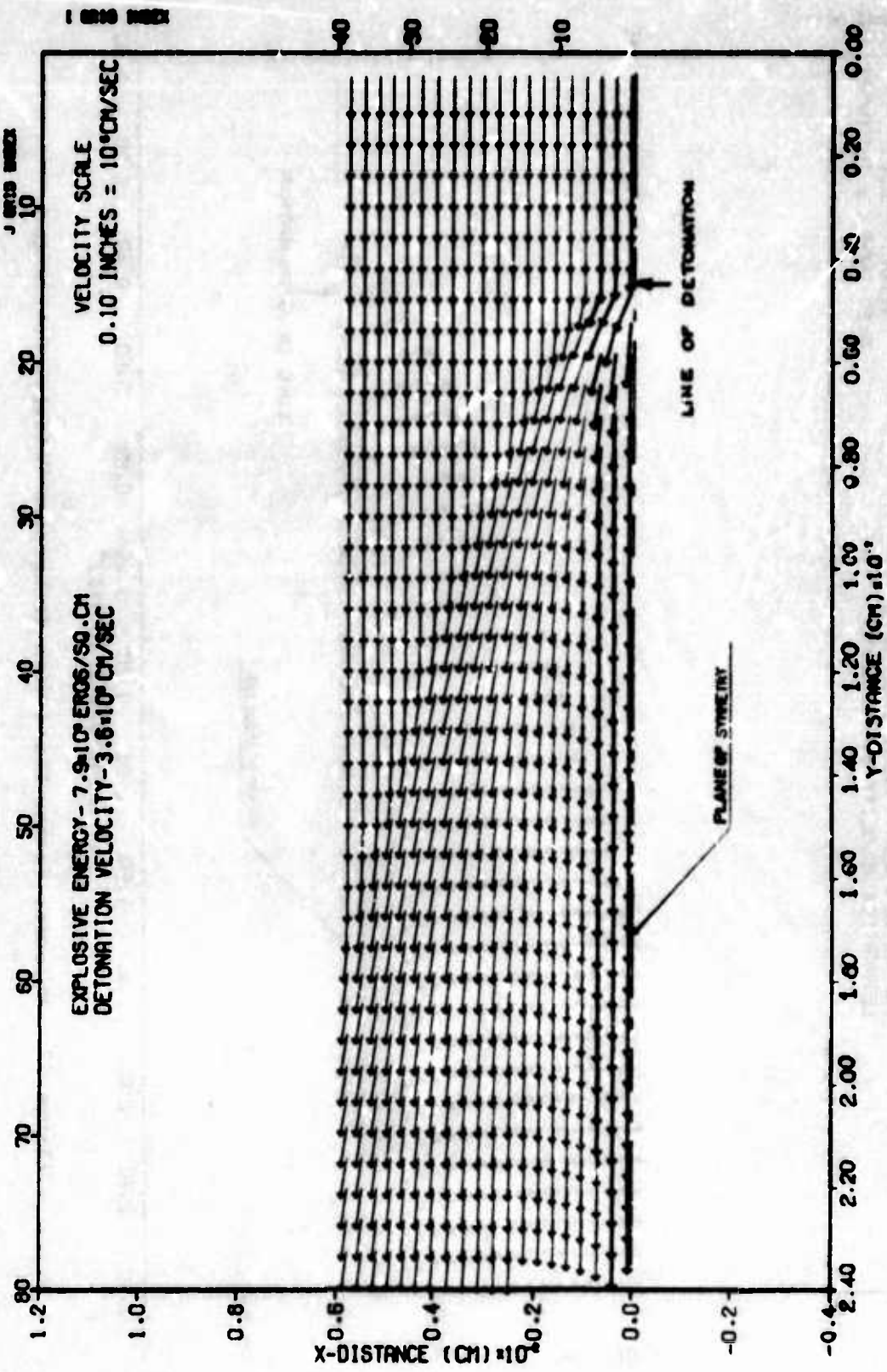


Figure 11. Velocity Vector Field

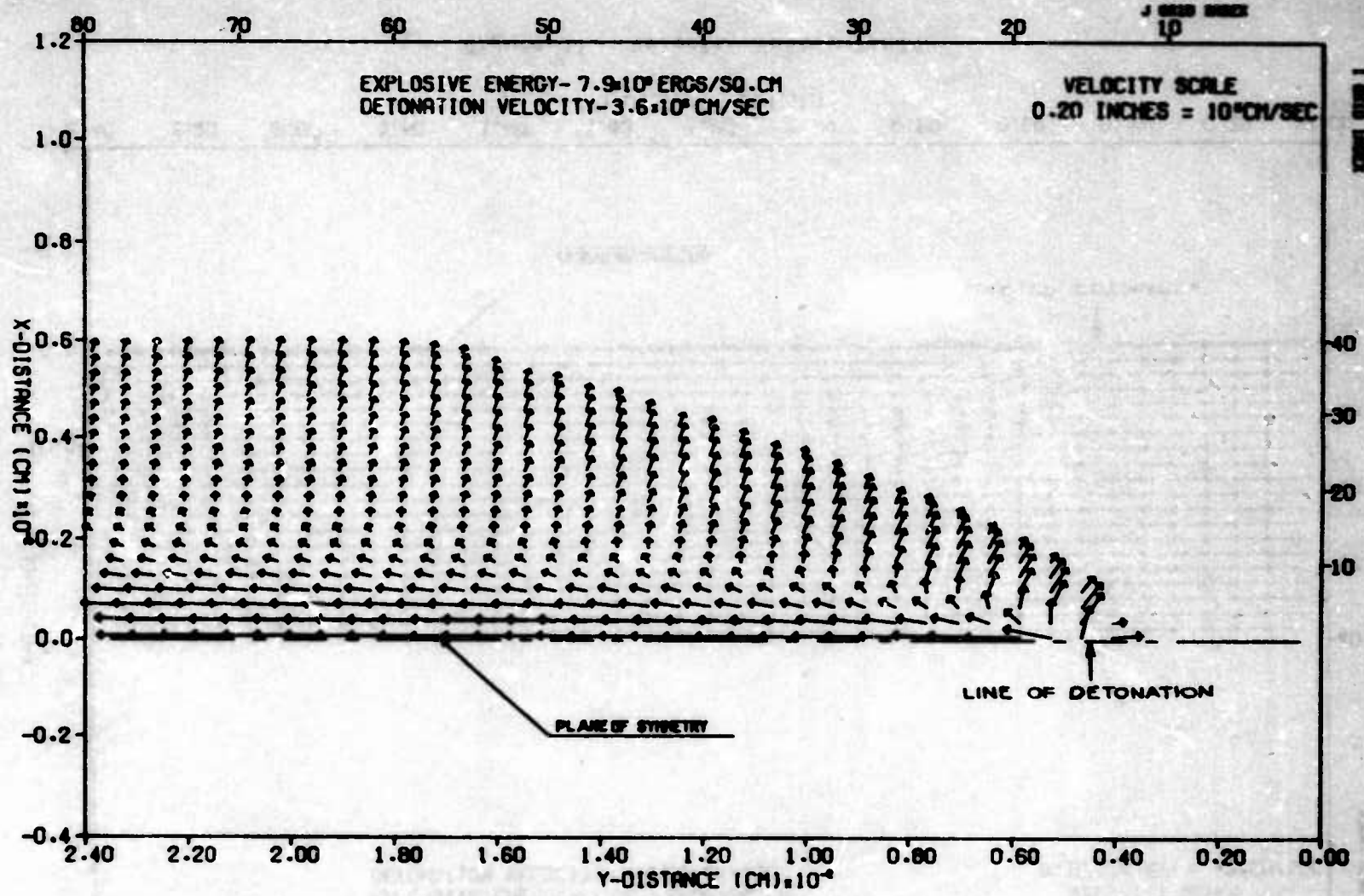


Figure 12. Velocity Vector Field With Background Velocity Subtracted

The solution exceeds the upper grid boundary for Y distances greater than 180 cm. The small oscillations on the pressure, energy and density contours (Figures 8 through 10) are due to the contouring scheme in the SHPLOTL code and do not represent oscillations in the flow field. The disturbance of the uniform flow field is apparent in the velocity vector field of Figure 11. By subtracting the background velocity (uniform flow velocity) one can see in Figure 12 the instantaneous velocity field produced by a line source moving along the plane through still gas. This vector field corresponds to a change of coordinates back to a fixed sheet of explosives with a detonation moving along the sheet.

The shock front locations for the four energy values, all at $3.6 \cdot 10^5$ cm/sec, are superimposed in Figure 13. We observe that the shock forms out in front of the line of detonation in the incoming uniform flow and is normal to the plane of symmetry of the problem. The shock front bends quite dramatically in its departure from the plane of symmetry moving backwards past the line of detonation giving a shock front shape not unlike that formed by a blunt body in hypersonic flow (33). The shock then tends to curve less dramatically, approaching the Mach angle for the free stream hypersonic flow at large distances downstream from the point of detonation. The Mach angle, μ, is defined by Reference 39

$$\sin\mu = \frac{1}{M} \qquad (22)$$

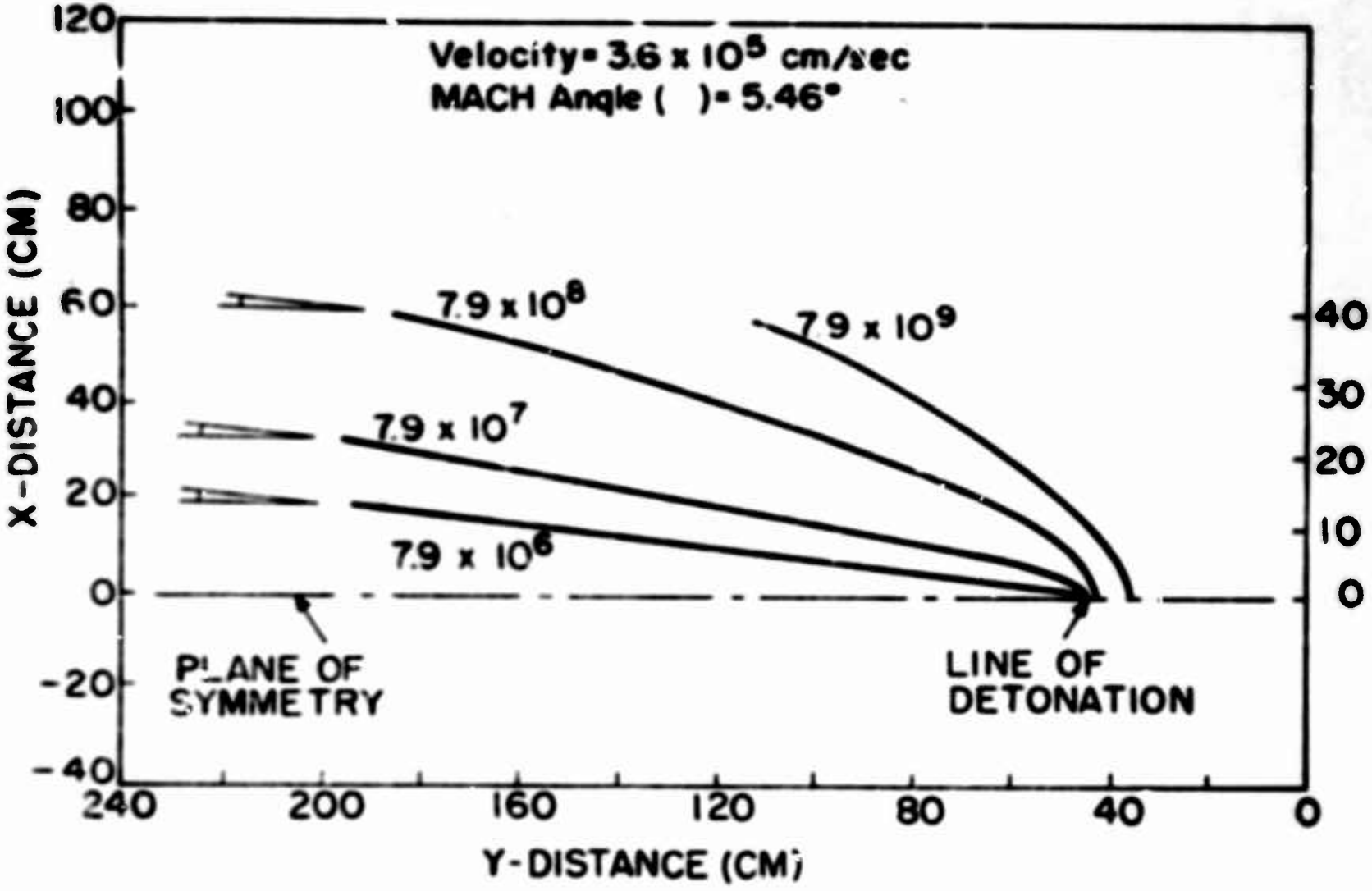


Figure 13. Numerical Solution for Four Energy Values

where M is the Mach number of the free stream flow. The shock front locations for three detonation velocities, all at the energy level 7.9×10^7 ergs/cm^2, are superimposed in Figure 14. One can observe that a slower hypersonic flow velocity (velocity of detonation) results in a slight shift of the shock upstream and a fuller shock profile.

EMPIRICAL FIT TO NUMERICAL SOLUTION

Several approaches to finding an empirical equation for shock location curves were tried. An attempt at normalizing the numerically obtained curves to find a single curve normalized by energy and velocity was not successful. Polynomials and exponentials did not prove satisfactory. A reasonably satisfactory fit was obtained using

$$Y = A + BX^n \tag{23}$$

where A, B and n are functions of the energy and velocity of detonation. Solution of this relationship for the numerically obtained data points provides the following functional relationships:

$$\begin{aligned} A &= -\left(\frac{E}{\alpha}\right)^{\beta} \\ B &= 10(\gamma - \theta \log_{10} E) \\ n &= \zeta + \phi \log_{10}\left(\frac{E}{10^{\zeta}}\right) \end{aligned} \tag{24}$$

where

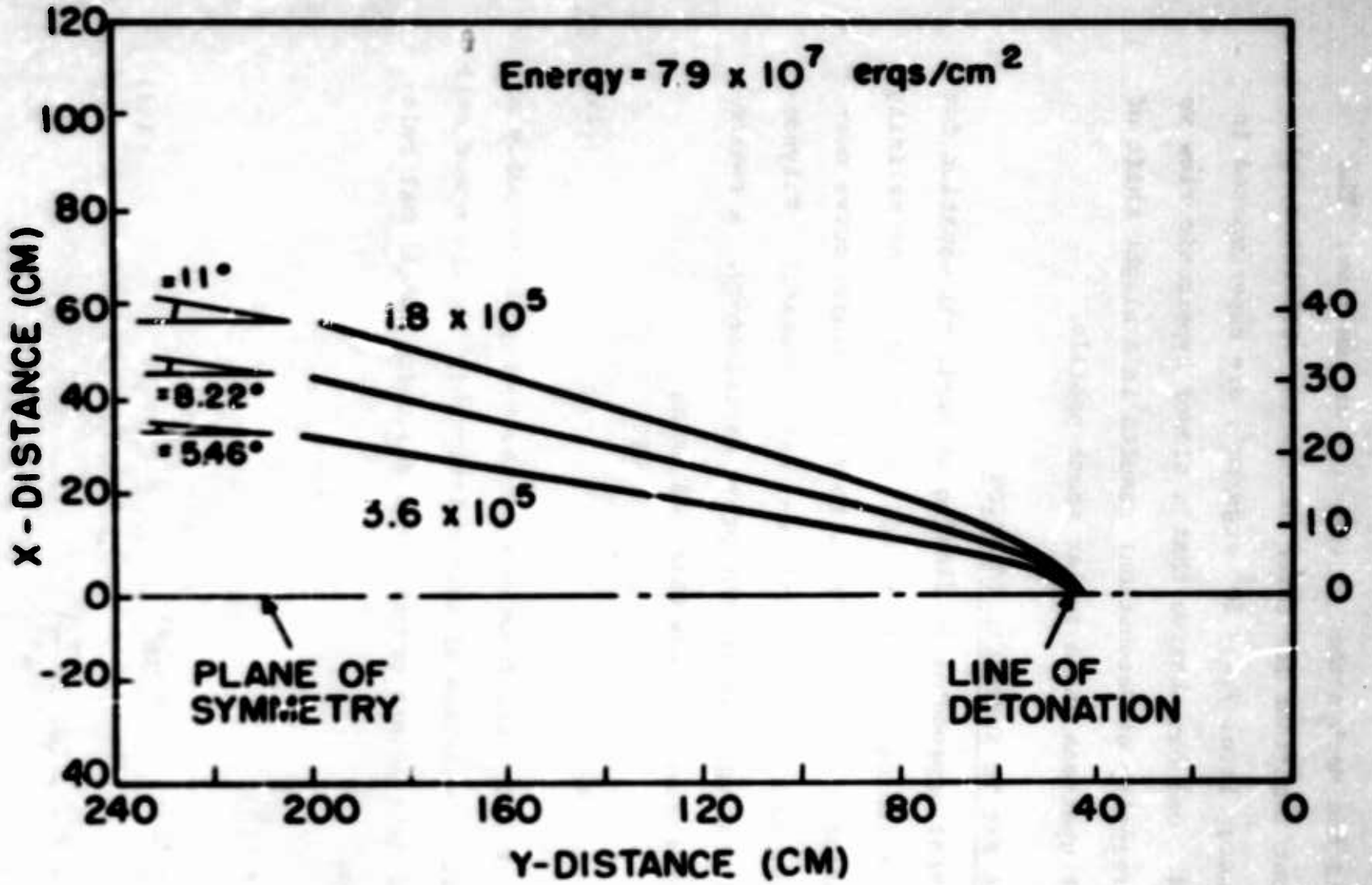


Figure 14. Numerical Solution for Three Velocity Values

E = detonation energy (ergs/cm^2)

V = detonation velocity (cm/sec)

and

$$\alpha = 30.45V - 3.175 \times 10^{6}$$

$$\beta = 0.368 - 3.28 \times 10^{-7}V$$

$$\gamma = 1.8 + V \times 10^{-5}$$

$$\theta = 0.315 + V \times 10^{-6}$$

$$\xi = 1.36 - 9.0 \times 10^{-7}V$$

$$\phi = 5.12 \times 10^{-7}V$$

A comparison of the numerically obtained curves with this empirical fit is shown in Figure 15.

Additional plots from solution of the idealized (energy-only) problem for other values of energy and velocity of detonation are provided in Appendix B.

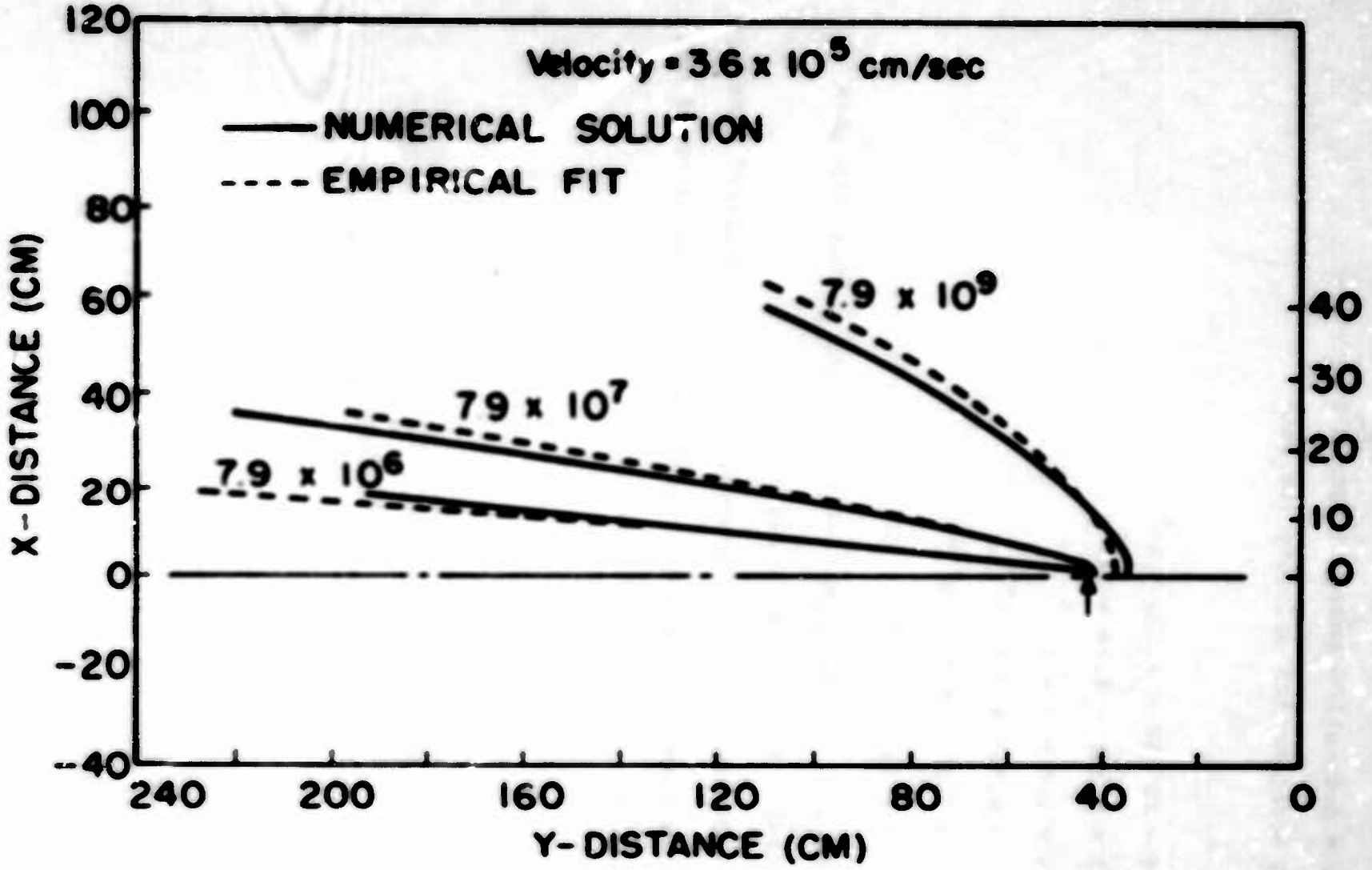


Figure 13. Comparison of Numerical Solution and Empirical Fit

SECTION III

MASS AND ENERGY ADDITION

The results obtained in the previous section are obviously limited by the idealizing assumptions used. A more realistic solution is obtained by recognizing that in the detonation of an explosive material, mass in the form of detonation products is released as well as energy. A further improvement is obtained by treating the gases in the surrounding medium and the gaseous detonation products as ideal gases while recognizing the ratio of specific heats (γ) for each gas as being a temperature-dependent parameter. A method of accounting for several distinct mass species is required in anticipation of the post-detonation combustion problem addressed in the next section. Obviously an analytical solution is not possible and a more sophisticated numerical approach is required.

MULTI-MATERIAL HYDRODYNAMIC CODE

In developing a computer code suitable for the solution of the real sheet-explosion problem, the first order of business is to determine how many different mass species must be accounted for. A review of the detonation products from explosive materials (34) indicates that TNT, one of the most common explosive materials, is also one of the most oxygen negative explosives (i.e., detonation products contain a large amount of combustible materials) and thus should provide a substantial post-detonation combustion effect. The

quantity of each specie of detonation product produced is dependent upon the initial density of the explosive material. An initial density of 1.47 gr/cm^3 was arbitrarily selected resulting in the release of the nine species of detonation products listed in Table I.

TABLE I

Detonation Products from TNT
(moles of each/kg of TNT)

CO - 1.8	N_2 - 5.9	CH_3OH - 4.29
CO_2 - 9.5	CH_4 - 0.1	C(s) - 14.2
H_2O - 1.3	NH_3 - 0.4	HCN - 1.1

By including the oxygen from the surrounding air, ten mass species are involved in the problem (air considered as containing only oxygen and nitrogen). At the detonation temperature (approximately 4000 °K) substantial dissociation occurs, but a short distance from the detonation point the temperature is markedly reduced and much recombination has occurred. It is therefore a reasonable approximation to assume no dissociation throughout the entire blast-wave problem. The quantities of CH_4 and NH_3 are so small that they may be ignored as separate species for which accountability must be maintained. However, the mass associated with the CH_4 and NH_3 must be accounted for. A way of accomplishing this is to assume that these two species dissociate into carbon, hydrogen, and nitrogen, and increase the quantities of each of these species an appropriate amount so that the total mass

remains constant. Thus, the following increase in the quantities of carbon, hydrogen, and nitrogen is made to account for the ignored CH_4 and NH_3:

C - 0.1 moles/kg of TNT

H_2 - 0.8 moles/kg of TNT

N_2 - 0.2 moles/kg of TNT

By using a similar approach, a further reduction in the number of mass species can be obtained by considering the equilibrium curves (41) for HCN and CH_3OH (Figure 16). At temperatures below 4000 °K these materials dissociate into H_2, N_2, CO and solid carbon. Realizing that the temperature will be much less than 4000 °K in all but one or two cells, it is reasonable to assume that this dissociation has occurred at all locations in the flow field. The amounts of H_2, N_2, CO, and C(s) are increased an appropriate amount to account for the dissociation of HCN and CH_3OH; thus keeping the total mass constant.

HCN: H_2 - 0.55 moles/kg of TNT

N_2 - 0.55 moles/kg of TNT

C - 1.1 moles/kg of TNT

CH_3OH: CO - 4.29 moles/kg of TNT

H_2 - 8.58 moles/kg of TNT

The total number of species to be accounted for can then be reduced to the seven listed in Table II.

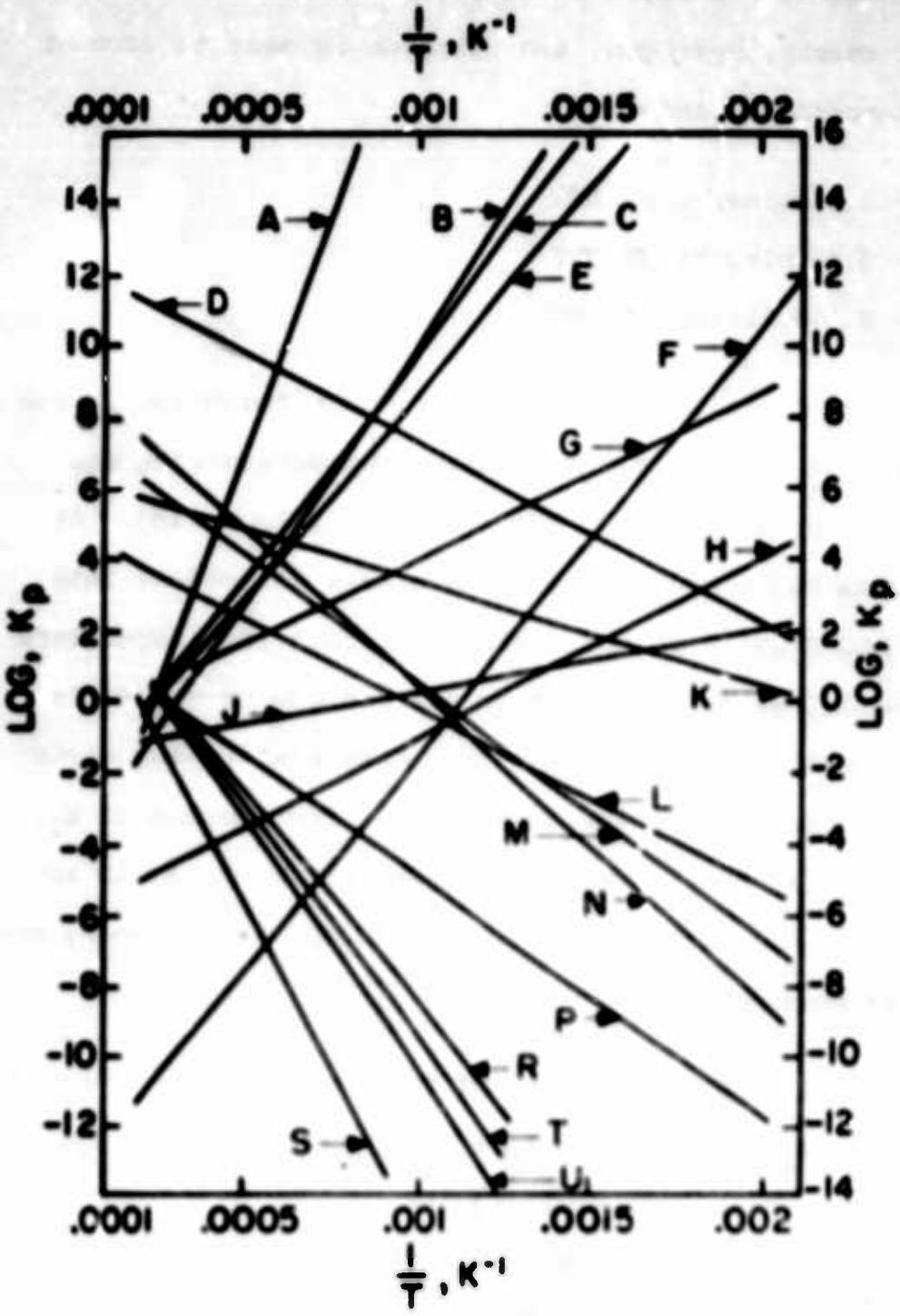


Figure 16. Equilibrium Constant versus Temperature, K

See Key, page 45.

KEY:

A $N + O \rightarrow NO$

B $CO + \frac{1}{2} O_2 \rightarrow CO_2$

C $H_2 + \frac{1}{2} O_2 \rightarrow H_2O$

D $CH_3OH \rightarrow CO + 2H_2$

E $C_2H_2 \rightarrow 2C + H_2$

F $CO + 3H_2 \rightarrow CH_4 + H_2O$

G $NO \rightarrow \frac{1}{2} N_2 + \frac{1}{2} O_2$

H $C + 2H_2 \rightarrow CH_4$

J $CO + H_2O \rightarrow H_2 + CO_2$

K $NH_3 \rightarrow \frac{1}{2} N_2 + \frac{1}{2} H_2$

L $SO_3 \rightarrow SO_2 + \frac{1}{2} O_2$

M $C + H_2O \rightarrow H_2 + CO$

N $2CO \rightarrow CO_2 + C$

P $H_2 + 2C + N_2 \rightarrow 2HCN$

R $\frac{1}{2} H_2 \rightarrow H$

S $\frac{1}{2} N_2 \rightarrow N$

T $\frac{1}{2} O_2 \rightarrow O$

U $H_2O \rightarrow \frac{1}{2} H_2 + OH$

TABLE II

Revised Material Accountability List
(moles/kg of TNT)

CO - 6.05	H_2O - 1.3	O_2 (background flow only)
CO_2 - 9.5	N_2 - 6.65	
H_2 - 9.93	C(s) - 15.4	

Thus, a seven-material hydrodynamic code could be used, within limitations, to approximate a more realistic solution of the blast wave formed by the explosion of a TNT sheet.

A multi-material version of the SHELL code, called RADISH 5, was already in existence at the start of this work (35). This code has the capability of determining the inviscid, compressible hydrodynamic behavior of five distinct mass species and includes effects of radiation diffusion. It is limited to axisymmetric coordinates as was SHELL OIL but includes temperature as another dependent variable for which accountability is maintained in each cell (in addition to pressure, mass, internal energy, and two components of velocity as in SHELL OIL).

In developing a computer code for the sheet explosion problem, RADISH 5 was used as the basic vehicle. The radiation diffusion portions were deleted and the code modified to operate in either axisymmetric or cartesian coordinates. It was expanded to accommodate seven materials, and a mass and energy deposition scheme was included. This revised code, titled SHELTC7, is the primary tool for use in this and the following section (see Appendix C for details of SHELTC7).

MULTI-MATERIAL EQUATION OF STATE

In using SHELTC7, a multi-material equation of state is required which will provide the pressure and temperature of the mixture of detonation products and air when the internal energy and quantity of each type of mass are known. Such an equation of state was developed, assuming the mixture to consist of six gases and solid carbon where the gases obey the ideal gas equation

$$P = \frac{nRT}{v} \tag{26}$$

where

P = pressure

n = number of moles

R = ideal gas constant

T = temperature

v = specific volume

and whose ratios of specific heats are functions of temperature.

$$\gamma = C_p/C_v = \gamma(T) \tag{27}$$

C_p and C_v data for the six gases listed in Table II are provided in the JANAF Thermochemical Tables (38) for temperature ranges of interest. C_v data for solid carbon are available in Reference 37. A polynomial fitting routine was used to obtain the expressions for C_p and C_v as functions of temperature listed in Table III. These C_p and C_v expressions were

TABLE III

Specific Heats for Six Gases and Solid Carbon

N_2 $\quad C_p = 6.393 + (1.7789 \times 10^{-3})T - (4.102 \times 10^{-7})T^2 + (3.1875 \times 10^{-11})T^3$

$\quad C_v = 4.4163 + (1.7675 \times 10^{-3})T - (4.034 \times 10^{-7})T^2 + (3.0933 \times 10^{-11})T^3$

O_2 $\quad C_p = 6.6046 + (1.8603 \times 10^{-3})T - (3.6125 \times 10^{-7})T^2 + (2.5902 \times 10^{-11})T^3$

$\quad C_v = 4.608 + (1.8774 \times 10^{-3})T - (3.6968 \times 10^{-7})T^2 + (2.6853 \times 10^{-11})T^3$

CO $\quad C_p = 6.4114 + (1.8637 \times 10^{-3})T - (4.4326 \times 10^{-7})T^2 + (3.5074 \times 10^{-11})T^3$

$\quad C_v = 4.4239 + (1.8638 \times 10^{-3})T - (4.4233 \times 10^{-7})T^2 + (3.4856 \times 10^{-11})T^3$

CO_2 $\quad C_p = 7.8268 + (5.9678 \times 10^{-3})T - (1.5678 \times 10^{-6})T^2 + (1.3213 \times 10^{-10})T^3$

$\quad C_v = 5.8423 + (5.9368 \times 10^{-3})T - (1.5217 \times 10^{-6})T^2 + (1.2629 \times 10^{-10})T^3$

H_2 $\quad C_p = 6.5548 + (8.6152 \times 10^{-4})T - (2.7312 \times 10^{-8})T^2 - (3.1196 \times 10^{-12})T^3$

$\quad C_v = 4.5507 + (9.0598 \times 10^{-4})T - (4.9424 \times 10^{-8})T^2 - (3.1041 \times 10^{-12})T^3$

H_2O $\quad C_p = 6.7245 + (3.9967 \times 10^{-3})T - (7.4975 \times 10^{-7})T^2 + (4.9546 \times 10^{-11})T^3$

$\quad C_v = 4.7832 + (3.9428 \times 10^{-3})T - (7.9342 \times 10^{-7})T^2 + (5.4627 \times 10^{-11})T^3$

Solid Carbon

$T < 1600$ °K $\quad \bar{C}_v = 0.8 + 0.002556T$

$T \geq 1600 \quad \bar{C}_v = 4.2 + 4.09 \times 10^{-4}T$

Units are $\frac{\text{calories}}{\text{gm-°K}}$

formed into computational subroutines to the main equation of state subroutine so that for any temperature, gamma can be computed for each mass specie.

$$\gamma(T) = \frac{C_p(T)}{C_v(T)} \tag{28}$$

The governing equations used in the equation of state subroutine are

$$P = (\gamma_i - 1)\frac{I_i}{v_i} \qquad (i = 1,2,\cdots,6) \tag{29}$$

$$P = n_i \frac{RT}{V_i} \qquad (i = 1,2,\cdots,6) \tag{30}$$

$$v_i = \frac{V_i}{m_i} \qquad (i = 1,2,\cdots,6) \tag{31}$$

$$P_i = n_i \frac{RT}{V} \qquad (i = 1,2,\cdots,6) \tag{32}$$

$$P = \sum_{i=1}^{6} P_i \qquad (i = 1,2,\cdots,6) \tag{33}$$

$$I = \sum_{i=1}^{6} m_i I_i \qquad (i = 1,2,\cdots,6) \tag{34}$$

$$m = \sum_{i=1}^{6} m_i \qquad (i = 1,2,\cdots 6,) \tag{35}$$

$$n_i = \frac{m_i}{M_i} \qquad (i = 1,2,\cdots,6) \tag{36}$$

where

P = cell pressure

P_i = partial pressure of ith gas specie

v_i = specific volume of ith gas specie

V = cell volume

V_i = partial volume of ith gas specie

I = total internal energy of cell

I_i = specific internal energy of ith gas specie

M = total cell mass

m_i = mass of ith specie

M_i = molecular weight of ith gas species

γ_i = specific heat ratio of ith gas specie

Combining equations 28, 29, and 30 to eliminate $P V_i$ we obtain

$$I_i = \frac{n_i RT}{(\gamma_i - 1) m_i} \tag{37}$$

Using this in equation 33 along with equation 35 we obtain

$$I = \sum_{i=1}^{6} I_i m_i = \sum_{i=1}^{6} \frac{n_i m_i RT}{(\gamma_i - 1) m_i} \tag{38}$$

$$I = RT \sum_{i=1}^{6} \frac{m_i}{M_i (\gamma_i - 1)} \tag{39}$$

Finally

$$T = \frac{I}{R \sum_{i=1}^{6} \frac{m_i}{M_i (\gamma_i - 1)}} \tag{40}$$

The temperature being found, the partial pressure for each specie is found by

$$P_i = \frac{n_i RT}{V} \qquad (i = 1,2,\cdots,6) \tag{41}$$

then,

$$P = \sum_{i=1}^{6} P_i \qquad (i = 1,2,\cdots,6) \tag{42}$$

provides the cell pressure.

The computation of the pressure and temperature using the multi-material equation of state subroutine (called STATE 7) is accomplished on a cell-by-cell basis throughout the entire grid, once during each time step. In each cell the volume of solid carbon is computed and, if significant, subtracted from the volume of the cell. Then the specific heat ratio (γ) is computed for the six gases at the present cell temperature using the formulations in Table III. The denominator for equation 39 is computed and a new temperature for the gas mixture is determined. Then the gas mixture and the solid carbon are brought into temperature equilibrium to obtain a new cell temperature. With the new cell temperature new gammas are computed and the above process repeated until the temperature converges. When temperature convergence has been obtained, the partial pressure for each gas specie is computed using equation 40. Finally the partial pressures are summed to obtain the new cell pressure.

THE COMPUTATIONAL PROBLEM

From previous experience with the idealized problem (point source in hypersonic flow), an energy value of 7.9×10^8 ergs/cm^2 was chosen as one not requiring excessive computer running time and that still could provide a substantial shock wave. From this energy value a thickness for the explosive sheet was determined. Using the following values

density = 1.47 gr/cm^3

energy = 1.13 kcals/gr

conversion = 4.186×10^{10} ergs/kcal

a thickness of 1.136×10^{-2} centimeters is obtained. Note that this is the thickness of material deposited above the axis of symmetry and corresponds to an actual sheet thickness of 2.272×10^{-2} centimeters.

With the thickness known the amount of explosive material (W) deposited into the computational grid per unit length of detonation travel can be determined

$$W = (\text{density}) \cdot (\text{thickness}) \cdot (\text{length}) \cdot (\text{width})$$

$$= (1.47)(1.136 \times 10^{-2})(1)(1)$$

$$W = 1.67 \times 10^{-2} \text{ gr/cm} \qquad (43)$$

Returning to the revised list of detonation products (Table II) and using the molecular weights assigned in Table IV, the amount of each specie provided by the detonation of a unit length of explosive can be determined.

TABLE IV

Molecular Weights of Mass Species

CO - 28	H_2 - 2	H_2O - 18
CO_2 - 44	O_2 - 32	N_2 - 28
C - 12		

$$\text{Amount of Each Specie Deposited (per unit length)} = \begin{pmatrix}\text{Total Mass Deposited Per Unit Length}\end{pmatrix} \cdot \begin{pmatrix}\text{Moles of Each Specie Per Unit Total Mass}\end{pmatrix} \cdot \begin{pmatrix}\text{Mole Weight of Specie}\end{pmatrix}$$

The amount of each specie that is deposited by a detonation of a unit length (cm) of explosive is listed in Table V.

TABLE V

Mass of Each Specie Deposited (gr/cm)

CO - 2.85×10^{-3}	H_2 - 3.32×10^{-4}
CO_2 - 6.98×10^{-3}	H_2O - 3.91×10^{-4}
C - 3.09×10^{-3}	N_2 - 3.11×10^{-3}

The computational grid was initially loaded with a uniform mixture of N_2 (75.5%) and O_2 (24.5%) representing air with an ambient density of 1.2×10^{-3} gr/cm^3, temperature of 292 °K and pressure of 14.7 psi. A modified CLAMTC code was used to establish and load this grid and to generate the zero cycle on the problem tape. The specific internal energy for each of the detonation products and the background flow gases were computed for ambient temperature using

$$I = \phi \frac{RT}{M} \tag{44}$$

where

$\phi \simeq 5/2$ for diatomic gases

$\phi = 3$ for triatomic gases

and are listed in Table VI.

TABLE VI

Internal Energy of Detonation Products
(ambient temperature) (ergs/gr)

N_2 - 2.165×10^9	CO - 2.095×10^9	H_2 - 29.3×10^9
O_2 - 1.895×10^9	CO_2 - 1.656×10^9	C(s) - 0
	H_2O - 4.05×10^9	Air - 2.0×10^9

The explosive detonation energy deposited with each deposition of mass was 4.72×10^{10} ergs per gram.

A background flow regeneration scheme (used to simulate incoming flow conditions) was devised and incorporated into the subroutine CDT of SHELTC7. The initial background flow conditions are reestablished in the first two rows of the grid (right edge - Figure 4) with appropriate adjustments to total problem mass, total problem energy and the mass of each species.

A mass and energy deposition scheme was devised and incorporated into subroutine CDT of the SHELTC7 code. A letter designator was assigned to each mass specie as listed in Table VII and used to account for that specie throughout the computation.

TABLE VII

Letter Designators for Mass Species

A - N_2	D - CO_2	G - C(s)
B - O_2	E - H_2O	
C - CO	F - H_2	

The length of material detonated (which establishes the amount of mass deposited) is determined by the time step and the detonation velocity. A momentum balance is performed between the material in the detonation cell and the material deposited, and new cell velocities are determined. Conservation of energy then determines how much kinetic energy is converted to internal energy. The mass of each specie and the total mass in the detonation cell are adjusted by the amount of mass deposited, and the internal energy of the cell is increased by the amount of explosive energy deposited. Finally, the total mass and the total energy in the grid are increased by the amount of each deposited during the detonation sequence.

COMPUTED RESULTS

Two solutions were obtained by using this more sophisticated computational scheme. Energy values of 7.9×10^8 and 7.9×10^9 ergs/cm^2, both at a detonation velocity of 3.6×10^5 cm/sec, were used corresponding to previously obtained solutions for the idealized (energy addition only) problem. The corresponding half thicknesses for a TNT sheet are 1.136×10^{-2} and 1.136×10^{-1} centimeters, respectively.

These problems were run on the CDC-6600 computer and required several hours of computation time for each.

The plotting code (SHPLOTL) was modified to accommodate the multi-material results obtained from the SHELTC7 computations and was then titled SHPLOT7. Plots of the temperature distribution (isotherms) are provided by this modified plotting code in addition to the plots of other variables (pressure, energy, velocity, etc.) mentioned previously in Section II.

The location in the x-y plane of the shock front, as determined by the SHELTC7 computation for the energy value of 7.9×10^8 ergs/cm^2 is shown in Figure 17. The shock again forms out in front of the line of detonation and then curves downstream, approaching the Mach angle at some distance downstream from the line of detonation. Contours of constant pressure (isobars) are shown in Figure 18. A high pressure zone occurs along the shock front with the highest pressures immediately adjacent to the line of detonation. In the region downstream from the line of detonation and along the plane of symmetry a low-pressure zone exists which causes a partial reversal of the initial outward flow of the mass. The inward-moving mass then converges on the plane of symmetry causing a new high-pressure region. Hence, high- and low-pressure regions alternate downstream along the plane of symmetry until the oscillations damp out and become insignificant. This same effect was observed in solutions of the idealized problem.

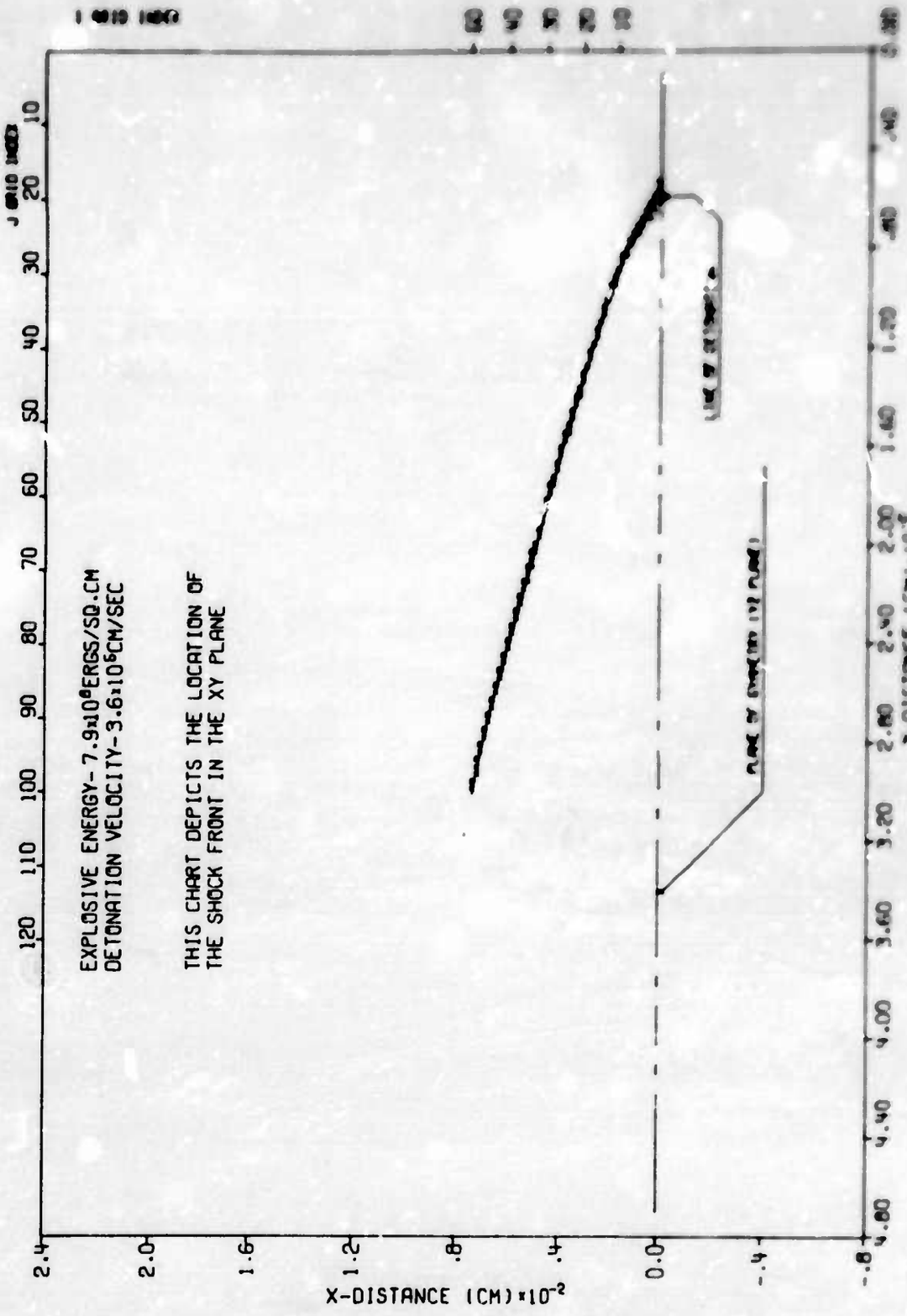


Figure 17. Shock Front Location in the XY Plane

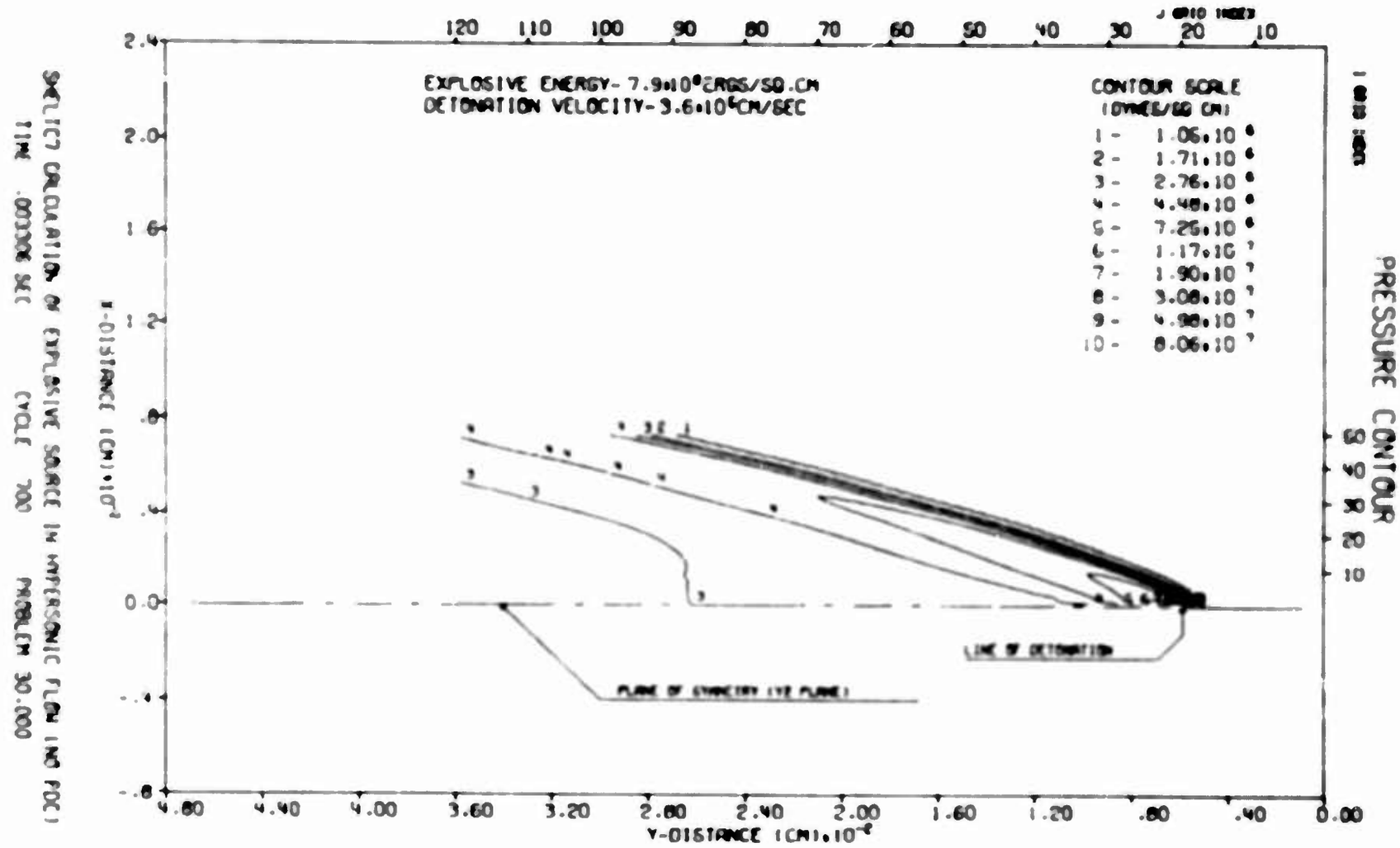


Figure 18. Constant Pressure Contours (Isobars)

Contours of constant energy are shown in Figure 19 and constant mass density in Figure 20. The concentration of a large percentage of the total mass involved in the blast wave in a narrow region close to the shock front can be seen. Figure 21 shows contours of constant temperature (isotherms) behind the blast wave. One may observe that the highest temperatures are located near the line of detonation and drop to below 3000 °K a short distance from that point. Hence, the assumption made earlier concerning the temperature distribution and the shift of chemical equilibrium is shown to be valid providing the time required for attaining equilibrium is short compared to the time it takes the mass to move from the line of detonation out to the cooler cells.

A comparison was made of the shock front locations determined from the idealized solution (energy addition only - Section II) and from the solution produced here (mass and energy addition). The comparison for the two energy levels, 7.9×10^8 and 7.9×10^9 ergs/cm^2, is shown in Figure 22. The shock front from the mass and energy addition solution forms slightly farther out in front of the line of detonation than the shock front from the idealized solution but it curves back more sharply. Thus, the shock front from the mass and energy addition solution crosses under the shock front from the idealized solution and continues beneath the idealized solution at all locations downstream from the line of detonation. In other words, the x-coordinate of the shock front from the mass and energy addition solution (X_{me}) is smaller

than the x-coordinate of the shock front from the idealized solution (X_i) by approximately the same amount for all values of y beyond the line of detonation.

$$X_i - X_{me} = \text{constant} \qquad \text{for } y > y_{detonation} \tag{45}$$

This compressed outline of the shock front from the mass and energy addition solution is caused by the detonation products having only y-momentum when they enter the flow field. In the transformed coordinate system used in solving these problems, the mass of the explosive material is moving in the same direction and at the same rate as the background gas. When this mass is injected into the background flow along the line of detonation (injected with no x-component of velocity), the same amount of detonation energy cannot accelerate and compress this increased mass as far in the x-direction as in the idealized (energy addition only) solution. Hence the shock front for the mass and energy addition solution will not have moved as far in the x-direction for any y-distance as will the shock front from the idealized solution.

No attempt was made to obtain an empirical fit for the mass and energy addition solution. Additional plots from mass and energy addition solution for other energy values and velocities of detonation are provided in Appendix D.

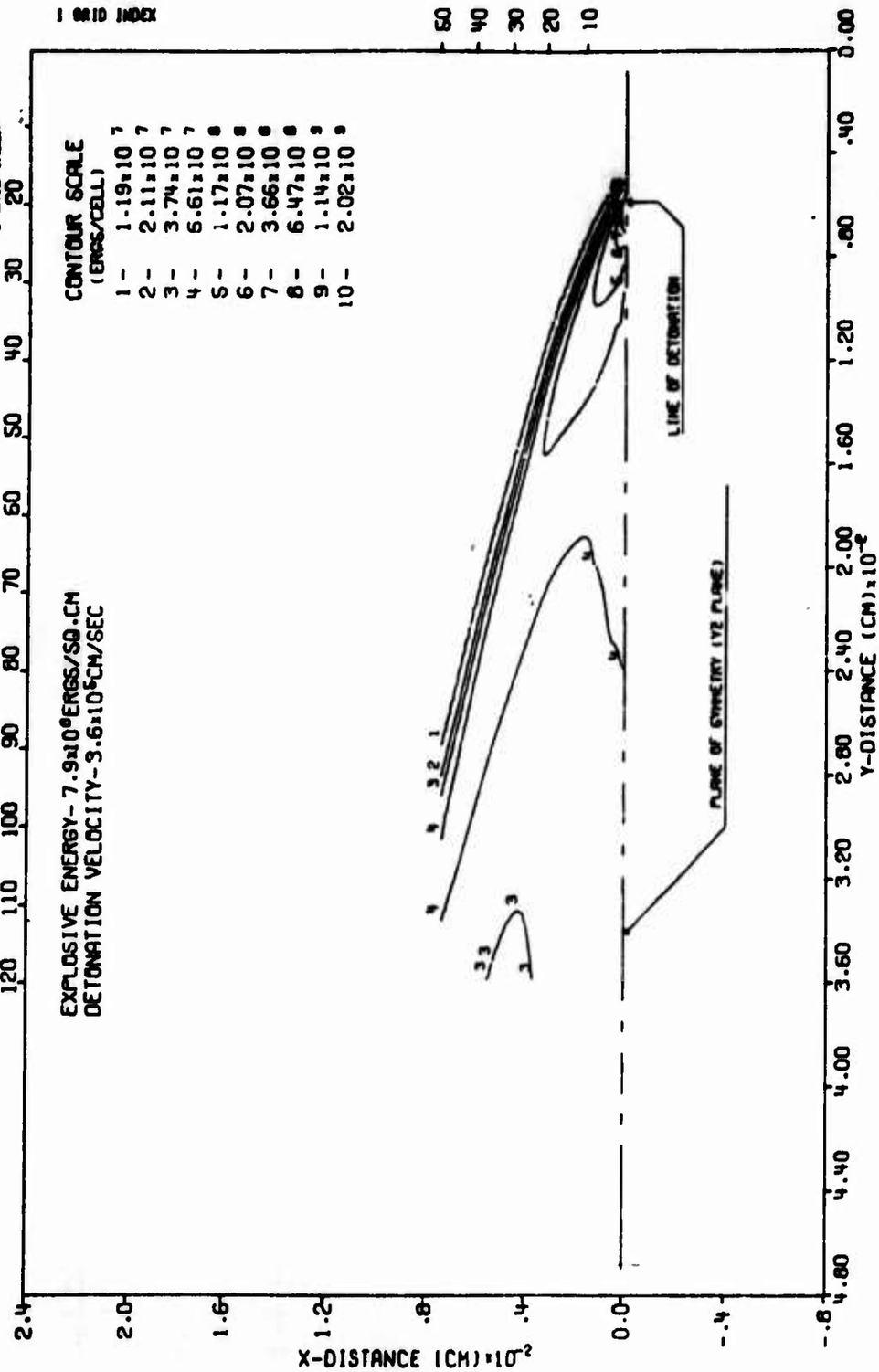


Figure 19. Constant Energy Contours

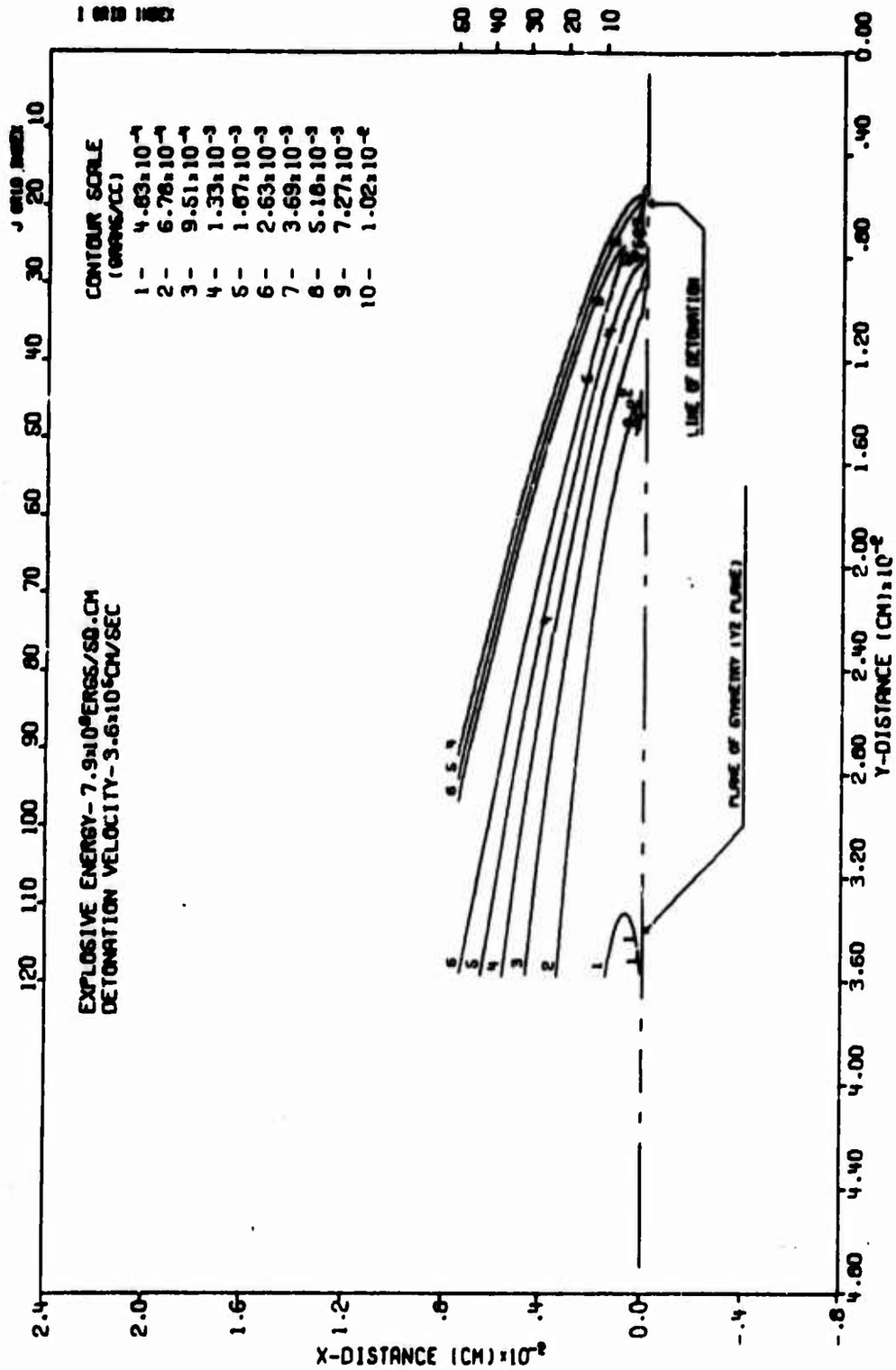


Figure 20. Constant Density Contour

SHELTC7 CALCULATION OF EXPLOSIVE SOURCE IN HYPERSONIC FLOW (NO PDC)
TIME .003306 SEC CYCLE 700 PROBLEM 30.000

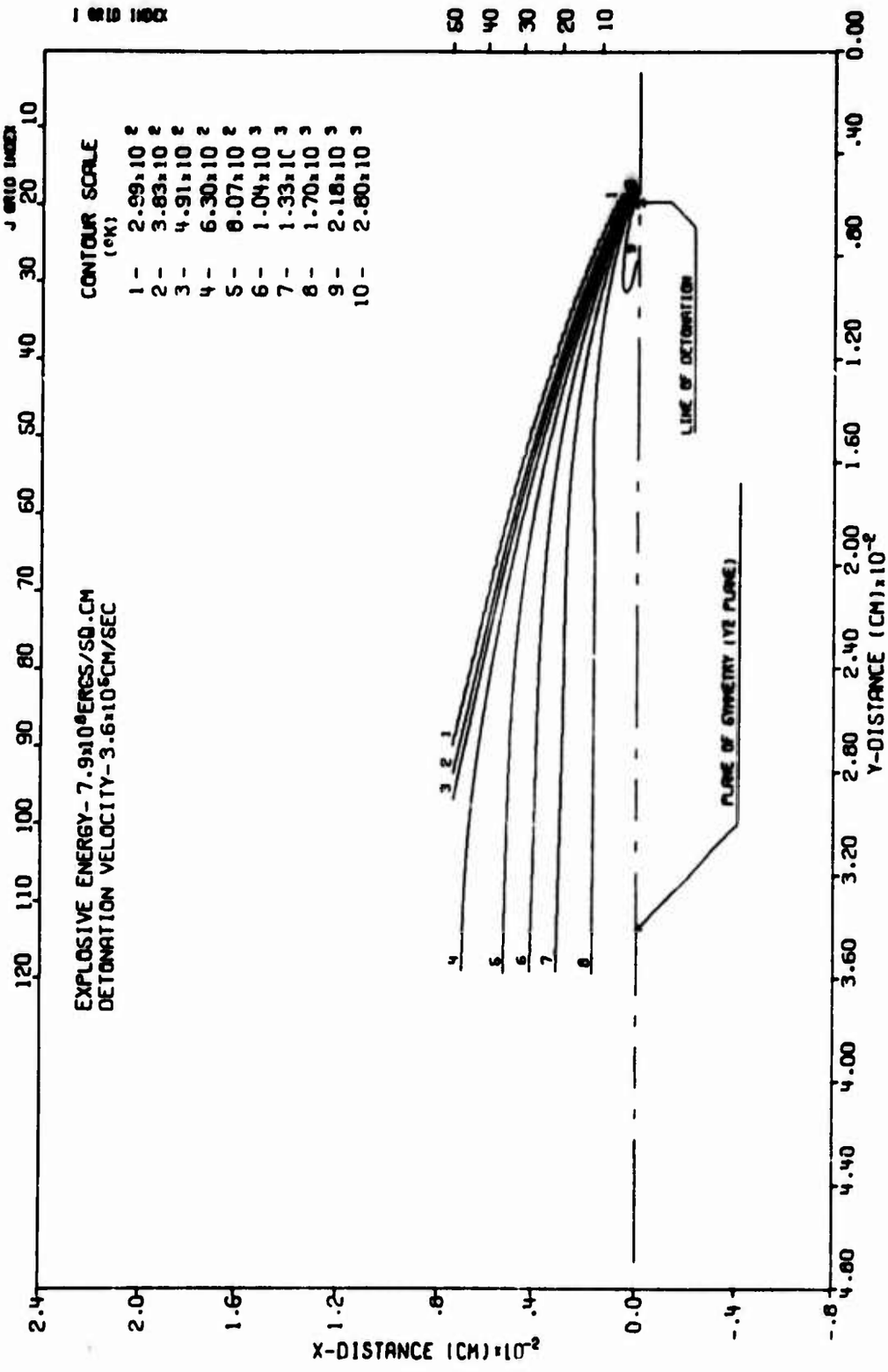


Figure 21. Constant Temperature Contours (Isotherms)

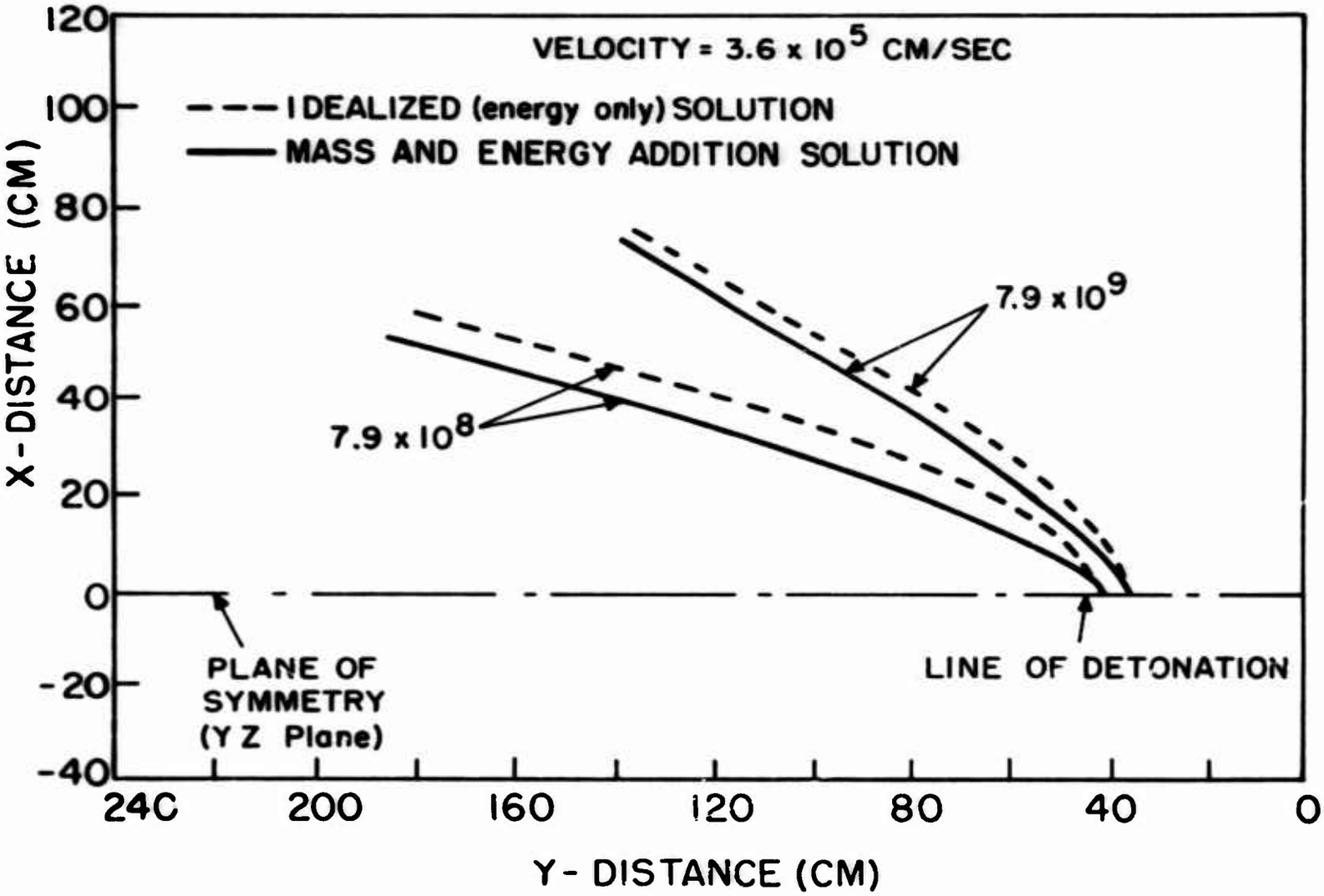


Figure 22. Comparison of Solutions (Idealized versus Mass and Energy Addition)

SECTION IV

POST-DETONATION COMBUSTION

Most explosive materials are highly oxygen negative. That is, in the basic detonation reaction (usually a decomposition of some complex molecule) all of the oxygen in the explosive material is consumed and substantial amounts of hot, combustible detonation products are produced. As these detonation products flow outward from the line of detonation, a material interface is formed between the detonation products and the surrounding medium. At first this interface is distinct with good material separation across the interface. The interface is not stable, however, and as outward flow continues, turbulent mixing and mass diffusion occur and an expanding zone of mixed materials (detonation products and surrounding air) is formed. Thus, the hot detonation products are provided with a source of oxygen, and combustion occurs to release an additional quantity of energy. This process is known as post-detonation combustion (PDC) or "after burning" and is a potential source of error for calculations where it is not included. Hence, the flow field behind the blast wave from the detonation of an oxygen negative explosive consists of (1) a region adjacent and downstream to the point of detonation filled only with products of detonation where no PDC occurs, (2) a mixed zone of detonation products and air where PDC occurs, and (3) an outlying zone of shocked air with no detonation products to be burned. This

distribution of materials through the different zones is depicted in Figure 23.

ESTIMATE OF PDC ENERGY RELEASE

An estimate of the total amount of energy which is available for release by PDC can be obtained by burning to completion all combustible detonation products in an abundance of oxygen. Using the detonation products from TNT listed in Table II, the following amounts of combustible materials are available:

C - 15.4 moles/kg of TNT

CO - 6.09 moles/kg of TNT

H_2 - 9.93 moles/kg of TNT

The combustion reactions are

$$1.\quad C + 1/2O_2 \overset{\Delta H_1}{\rightarrow} CO \tag{46}$$

$$2.\quad H_2 + 1/2O_2 \overset{\Delta H_2}{\rightarrow} H_2O \tag{47}$$

$$3.\quad CO + 1/2O_2 \overset{\Delta H_3}{\rightarrow} CO_2 \tag{48}$$

The heats of combustion for these reactions (36) are

ΔH_1 = -26,620 cals/mole of C

ΔH_2 = -68,320 cals/mole of H_2

ΔH_3 = -67,410 cals/mole of CO

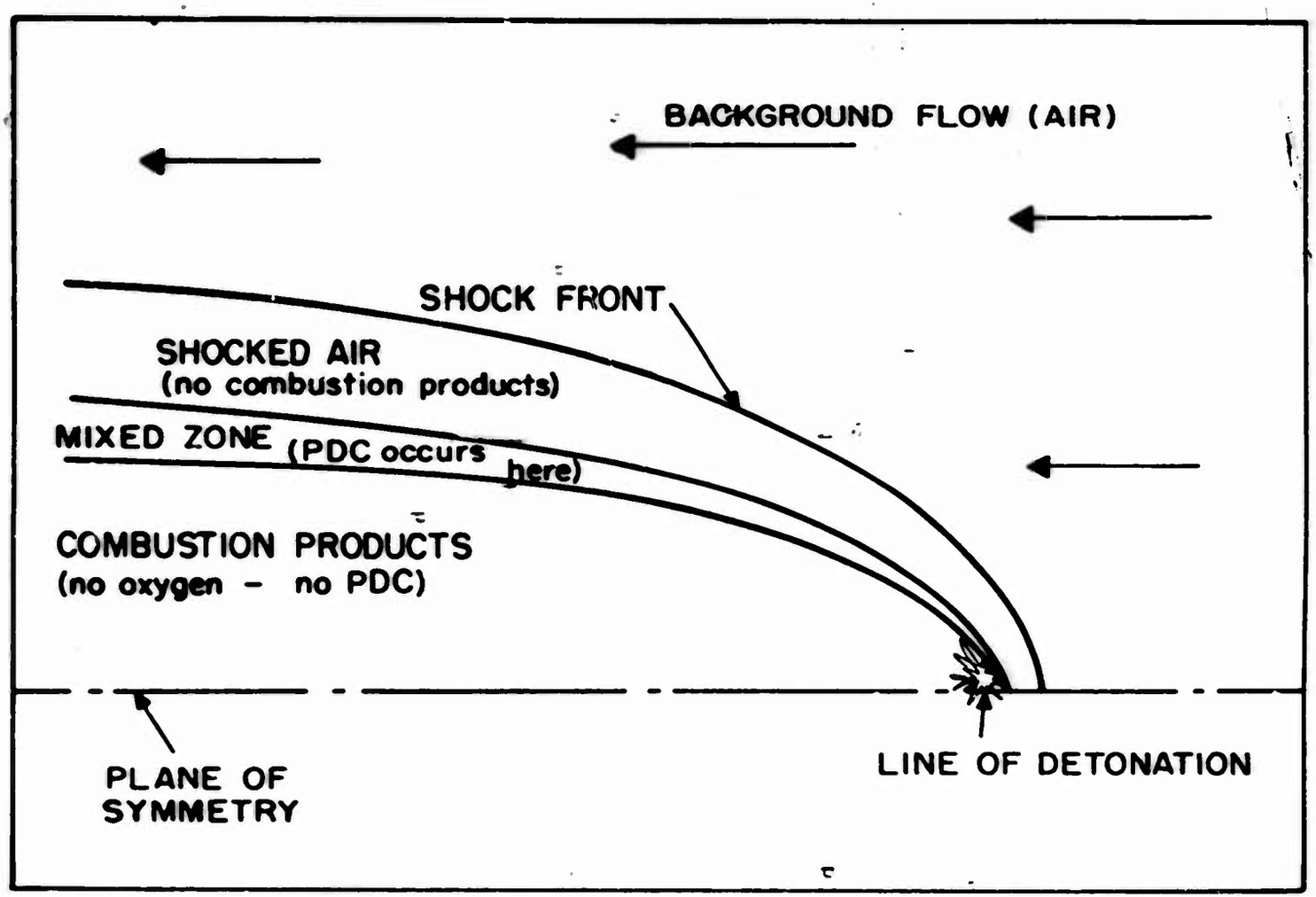


Figure 23. Material Zones behind the Blast Wave

The calculation of energy release, per kg of explosive material, follows:

$$\underline{C \rightarrow CO}$$

$$(15.4)(26{,}620) = 410{,}000 \text{ cals/kg} \tag{49}$$

$$\underline{H_2 \rightarrow H_2O}$$

$$(9.93)(68{,}320) = 678{,}000 \text{ cals/kg} \tag{50}$$

$$\underline{CO \rightarrow CO_2}$$

$$(6.09 + 15.4)(67{,}410) = 1{,}450{,}000 \text{ cals/kg} \tag{51}$$

The explosive energy released by a kg of TNT ($\rho = 1.47$) is (Reference 34)

$$Q_D = (1.13)\frac{\text{kcal}}{\text{gr}}(1000)\frac{\text{cal}}{\text{kcal}}(1000)\frac{\text{gr}}{\text{kg}}$$

$$= 1.13 \times 10^6 \frac{\text{cal}}{\text{kg}} \tag{52}$$

The ratio of PDC energy to detonation energy is

$$\text{ratio} = \frac{2.538 \times 10^6}{1.13 \times 10^6} = 2.245 \tag{53}$$

Thus, it would appear that more than twice as much energy can be produced by the PDC than is produced by the basic detonation of the TNT and that the contribution to the blast wave is substantial. Experimental evidence, however, indicated this is not true, because the detonation products are not all burned to completion and the burning which occurs does so over a long period of time. In fact, photographic evidence and post-detonation examination of large high-

explosive experiments indicate a substantial amount of unburned carbon is deposited around the vicinity of the detonation. The contribution of PDC to the early-time (under 10 milliseconds) formation of blast waves is not known and thus has been ignored in previous blast wave calculations. Some experimental evidence exists (43) which suggests that this may be a source of error, particularly at late times (several hundred milliseconds) after the detonation of a large explosive charge.

It was intended at the start of this study to develop an understanding of the post-detonation combustion process and to incorporate a combustion scheme into an existing hydrodynamic code so that some quantitative information on the role of PDC in the formation of a blast wave could be obtained without a several-year development of a completely new computer code. During the course of this work it has become apparent that this goal is difficult to achieve.

Post-detonation combustion is a complicated process involving mass transport, mixing of different mass species, and chemical reaction kinetics. The quantities of combustible materials and oxygen at any point in the flow field at any time depends both upon the overall hydrodynamic flow and upon the amount of mixing and diffusion which has occurred across the air-detonation product interface. There are several competing reactions occurring simultaneously. The reaction rate for a particular reaction depends upon the concentration of the reactants and upon a reaction rate coefficient which is,

in turn, dependent upon temperature. An exact determination of the PDC reaction requires the simultaneous solution of a series of differential rate equations of the form

$$\frac{d(x_i)}{dt} = (v_i'' - v_i')k \prod_{i=1}^{n} (x_i)^{v_i'} \qquad (i = 1,\cdots,n) \qquad (54)$$

where

x_i = molar concentration of i^{th} specie (moles/cm^3)

v_i'' = stoichiometric coefficient of i^{th} specie in products

v_i' = stoichiometric coefficient of i^{th} specie in reactants

k = reaction rate coefficient for the specified reaction

n = number of mass species

for each incremental volume (grid cell) in the flow field. Solving these equations for the three simplified combustion reactions used in determining the amount of PDC energy available (Equations 46, 47, and 48) will not provide correct results because the actual combustion process involves many other molecular and atomic species and several chemical reactions. Accurate results are obtained only if all of the reactions are included and the correct reaction rate constants are known. These equations have previously been solved for certain mixtures and computer programs for solving such reaction-rate problems have been developed (37). However, these programs depend upon iterative integration

schemes and require several seconds of computer running time to determine the progress of a reaction over some small period of time. To incorporate such an exact reaction rate subroutine into a hydrodynamic code (such as SHELTC7) would require the execution of that subroutine for every cell once during each cycle. Most blast-wave problems require several thousand grid cells and run several hundred cycles to obtain a solution. Hence, the computer time required for solving a hydrodynamic combustion problem would be several hundred hours on the fastest computers currently available. In addition, it would be necessary to maintain accountability of at least twelve distinct mass species in each cell of the computational grid (only seven species are stored in SHELTC7) which would greatly increase the amount of computer memory required.

A further difficulty was encountered in attempting to determine the mixing and diffusion which occur in the flow behind a blast wave. When study of the available hydrodynamic codes suitable for blast-wave calculations was started, it appeared that mixing and diffusion had been considered in their development and computation of the mixing in a multi-material flow field would not present a problem. During the course of the study it was discovered that such was not the case--all the codes suitable for two-dimensional blast-wave calculations had been originally developed for a single material and later adapted for multi-material operation. In the adaptation no means had been specifically provided for

computing the mixing and diffusion of one mass specie through another. Thus, it became apparent that no mathematically rigorous method for computing the mixing and diffusion of mass species which accompany the hydrodynamic flow behind a blast wave was available.

There was little hope, then, of performing an exact calculation of the PDC. The development of a new computer code to accomplish this would take several years and was beyond the scope of this study. It was discovered, however, during the study of SHELL hydrodynamic codes that mixing of the different mass species does occur in the multi-material versions of SHELL, even though no specific mechanism had been incorporated in the code to cause it to happen. This mixing, sometimes called "artificial diffusion," is inherent in all multi-material Eulerian codes and is an accidental rather than a planned occurrence.

With this discovery in mind it was decided to pursue the development of a scheme to approximate the contribution from PDC in a blast-wave calculation, realizing that experimental correlation would be required before good confidence in the approximation scheme could be obtained.

MIXING OF MASS SPECIES IN SHELTC7

Previous studies using the SHELL codes (16, 17) have referred to mass transport across a material interface, called an "artificial diffusion," as a disadvantage inherent in all Eulerian codes. However, for the approximation that is being developed it may be an advantage if satisfactory comparison

with experiment can be obtained. The basic equations from which SHELTC7 was developed (see Appendix A) are the equations for 2-D inviscid compressible flow and do not contain any terms which would provide mixing or diffusion. Hence, the code should produce only laminar flow. However, in the computational algorithm at the conclusion of the mass transport phase (subroutine Phase II in the code), all the mass species in any cell are considered to be uniformly distributed throughout the cell. This instantaneous dispersion or "diffusion" causes any new mass specie which has been introduced into the cell to be instantaneously transported across and throughout the cell. By this means, mass species are propagated from cell to cell and mixed with other species somewhat like the mixing and diffusion in real gases. But since this "effective mixing" does not result from the basic equations, it must be considered entirely artificial being caused directly by the computational scheme. Nevertheless, it does accomplish mixing and diffusion (superimposed upon the basic hydrodynamic flow), and may be useful much as the artificial or "effective viscosity" which is inherent in SHELL (see Appendix A for SHELL effective viscosity).

The mixing and diffusion in SHELTC7 occur only when there is a flow of mass and only in the direction of the flow velocity (i.e., when there is no component of velocity in a given direction, there is no mixing in that direction). The amount of mixing which occurs in a given time (i.e., the mixing rate) is governed by the magnitude of the flow velocity,

the concentration gradient of the various mass species, and the cell size of the computational grid. Expressed in another way, the time rate of change of the concentration of a mass specie in any direction is a function of the magnitude of the component of velocity in that direction and of the concentration gradient of that specie in that direction.

$$\frac{d(C_i)}{dt} = F\left(|V| , \frac{d(C_i)}{dR}\right) \qquad (i = 1, 2, \cdots, n)$$

where

(C_i) = mass concentration of i^{th} species (gm/vol)

V = component of velocity in specified direction (cm/sec)

R = distance in specified direction (cm)

t = time (sec)

n = number of mass species

The functional relationship has not been determined but is undoubtedly influenced by cell size. A literature search was conducted for a suitable experiment which could be calculated using SHELTC7, to permit comparison of the mixing in that code with experimental data. Recent work by McArtor (44) involving 2-D transverse injection of mass into supersonic flow offered hope but a preliminary single-material calculation using SHELLTC revealed that boundary layer effects dominate that experiment making it unsuitable for correlation with SHELTC7. Other experimental work (45, 46) was considered but none proved suitable for this purpose.

Realizing the need for a more detailed investigation and definition of the mixing in SHELTC7, if one accepts this mixing as a suitable approximation for the turbulent mixing and diffusion which occur during the propagation of a blast wave, then further work in the study of PDC can be accomplished. (Substantial discussion of the suitability of this approximation with others who are using and modifying multi-material versions of SHELL has, in general, resulted in agreement with the approximation.) It was decided that continuing the post-detonation combustion study using this approximation of material mixing would be meaningful, particularly if correlation with specifically designed experiments could later be obtained.

After some preliminary computer runs were attempted, it was discovered that an obvious excess of mixing was occurring near the line of detonation as a result of the mass deposition scheme incorporated into SHELTC7 (deposition of the detonation products into the flow field--described in Section III). In that scheme the amount of each of the mass species to be deposited is computed and then added to the mass of the cell immediately adjacent to the line of detonation. In so doing, each mass specie is instantaneously dispersed throughout the volume of that cell and thus is uniformly mixed with all the other mass species already in the cell. This results in an unrealistically large amount of mixing and precludes a sharp interface between the background flow of air and the incoming flow of detonation products. It can be argued that

the most distinct material interface in the entire flow field from an actual detonation will be found close to the line of detonation before significant mixing or diffusion has had time to occur. Thus, the mixing which occurs as a result of the mass-deposition scheme is not realistic and does not very well represent that which happens near an actual line detonation.

To correct this deficiency, a special subroutine called INTFACE was incorporated into SHELTC7 which creates a more distinct material interface near the line of detonation. INTFACE is executed once each cycle after the mass transport phase (subroutine Phase II) is completed. The subroutine begins in a cell adjacent to the plane of symmetry and a few cells upstream from the line of detonation (i = 1, j = 16 was chosen). The next cell adjacent to the beginning cell in the direction of increasing i index (i = 2, j = 16) is examined to see if it contains any detonation products. If it does, equal quantities of air (nitrogen and oxygen) and detonation products are exchanged between the two cells until either there is no air remaining in the first cell or no detonation products remaining in the second cell. The second cell is then designated as the new first cell and the next adjacent cell in the increasing i direction (i = 3) is designated the new second cell. The mass trading sequence is then repeated. This process is continued along the row (increasing i index) until cells with no detonation products are encountered. The j index is then increased by one and the

above sequence is repeated starting with the i index at the value of the last cell encountered in the previous sequence. The progression in the direction of increasing j index is continued for an arbitrary number of rows (14 was used). By this means the material interface between air and detonation products is reduced to essentially one or two cell widths for the first several rows near the line of detonation and the excess mixing associated with the mass deposition scheme is greatly reduced. Here again, experimental verification is required before real confidence in this interface preservation scheme can be attained.

The mass and energy addition solution for the energy value of 7.9×10^8 ergs/cm^2 and detonation velocity of 3.6×10^5 cm/sec (described in Section III) includes the distribution of the seven material species throughout the computational grid which results from the mixing which is inherent in SHELTC7 as influenced by subroutine INTFACE. The multimaterial plotting code SHPLOT7 was constructed to depict this distribution in two ways. First, contours linking locations where equal quantities of a selected specie exist are drawn for each mass specie as shown in Figures 24 through 30. The units of the contour scale are grams and refer to the number of grams of that specie in the cell at that location in the grid. The second method of depicting the mass distribution consists of assigning a symbol and a size for that symbol for increments of the mass fraction of the specie being plotted. The mass fraction of that specie in the cell is determined

at selected cells in a pattern throughout the grid. The appropriate symbol is then drawn (with the corresponding size) at the location of each of the selected cells. Thus, a graphic representation of how each mass specie is distributed through the grid is provided as shown in Figures 31 through 37. One can observe in Figures 24, 31 and 32 the distribution of N_2 and O_2 in the background flow (air) and within the blast wave. The concentration of O_2 within the blast wave is low, most of it having been deflected around the blast wave. The plot of constant mass contours for O_2, Figure 25, is not very satisfactory since extremely low concentrations of O_2 (associated with locations very near to the line of detonation) dominate that plot. Although the bulk of the N_2 from the background flow is deflected around the blast wave, the concentrations within the wave are somewhat higher than the amount which would correspond to the concentration of oxygen within the blast wave because N_2 is one of the products deposited from the detonation.

The distribution and concentration of the remaining five detonation products are shown in Figures 26 through 30 and 33 through 37. In particular, the distribution and concentration of CO_2 are shown quite clearly in Figure 34.

This approximate distribution of the material species within the blast wave is the foundation on which a post-detonation combustion scheme can be built.

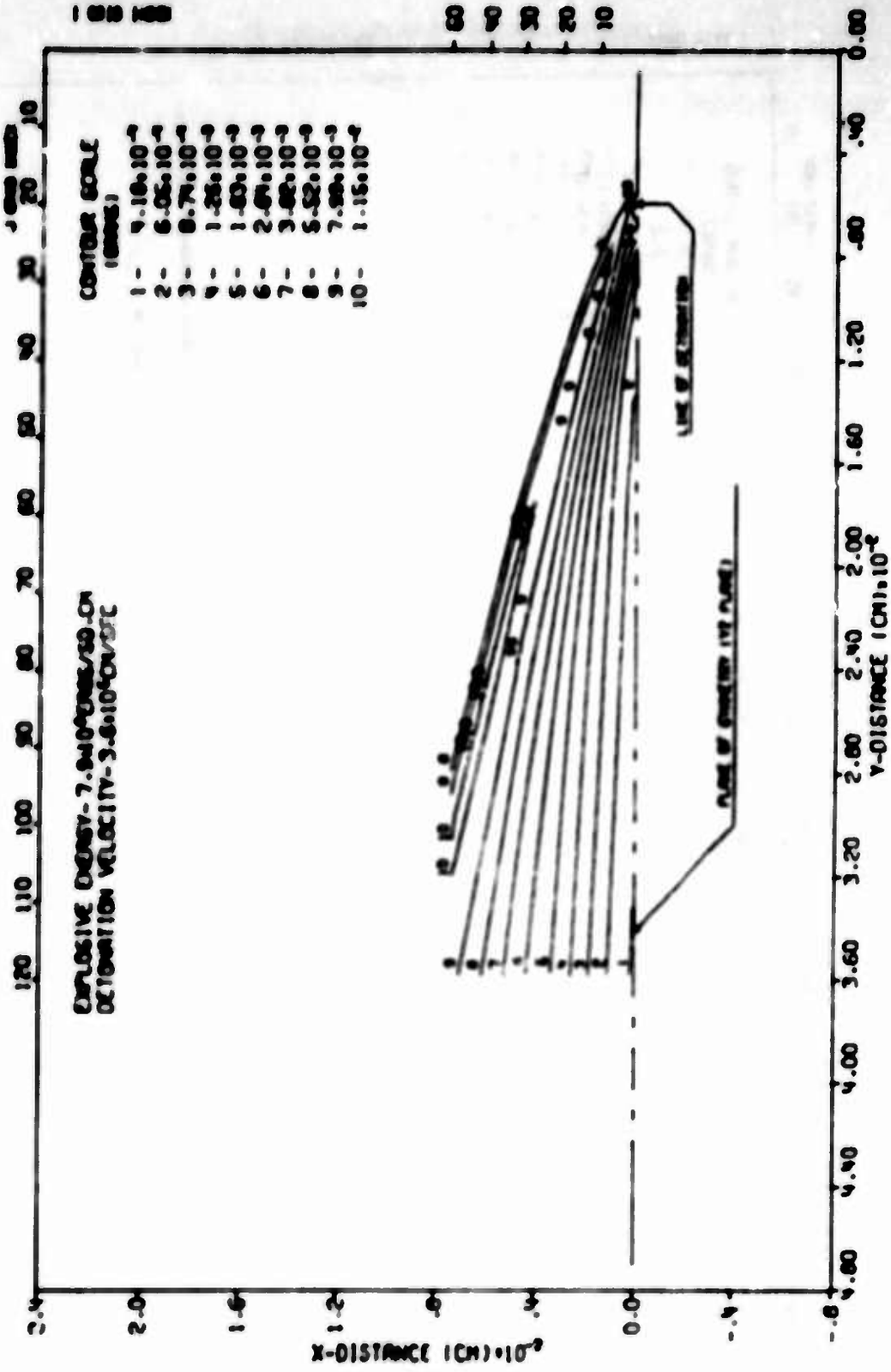


Figure 24. Constant Mass Contours--Species A (Nitrogen)

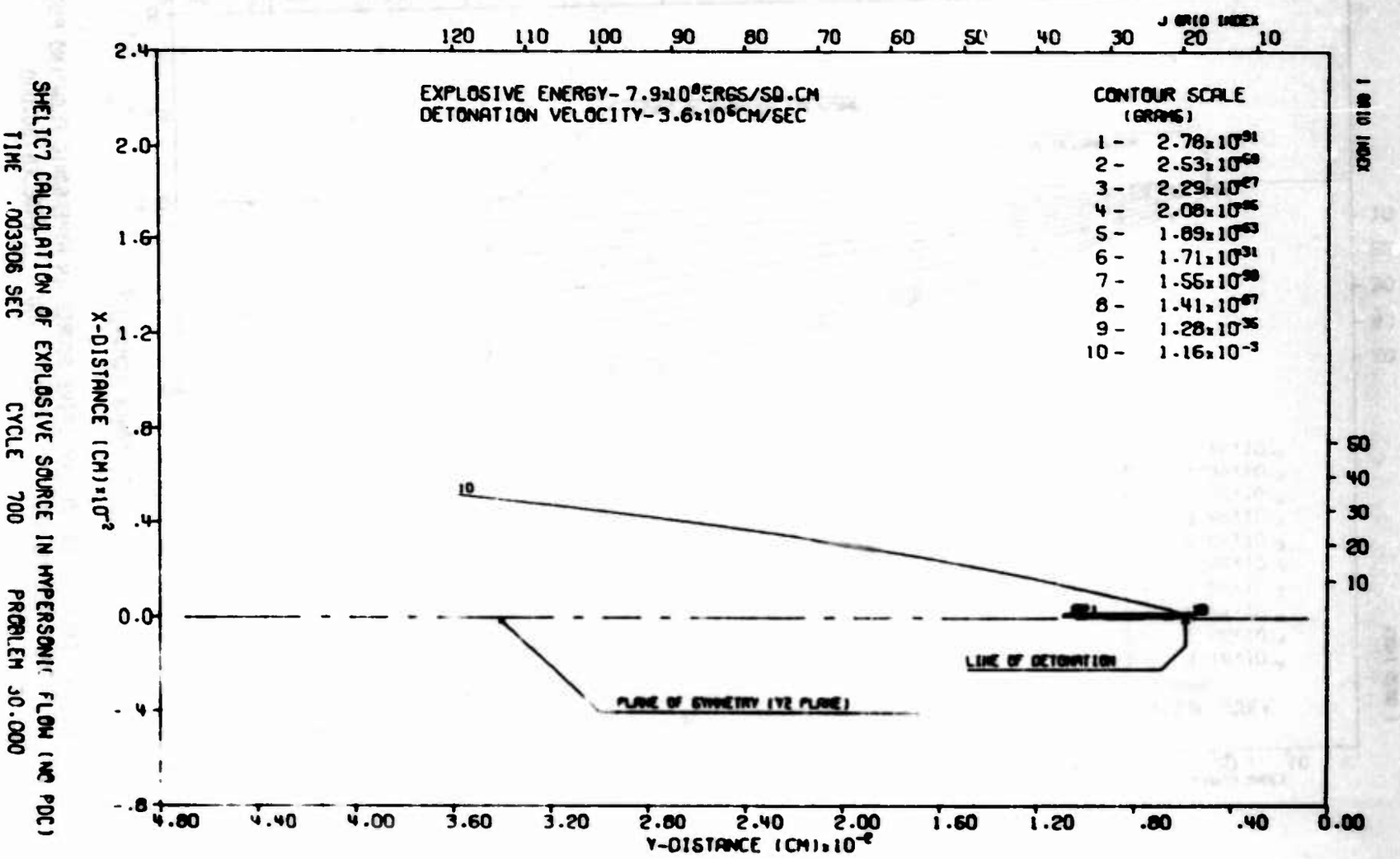


Figure 25. Constant Mass Contours--Species B (Oxygen)

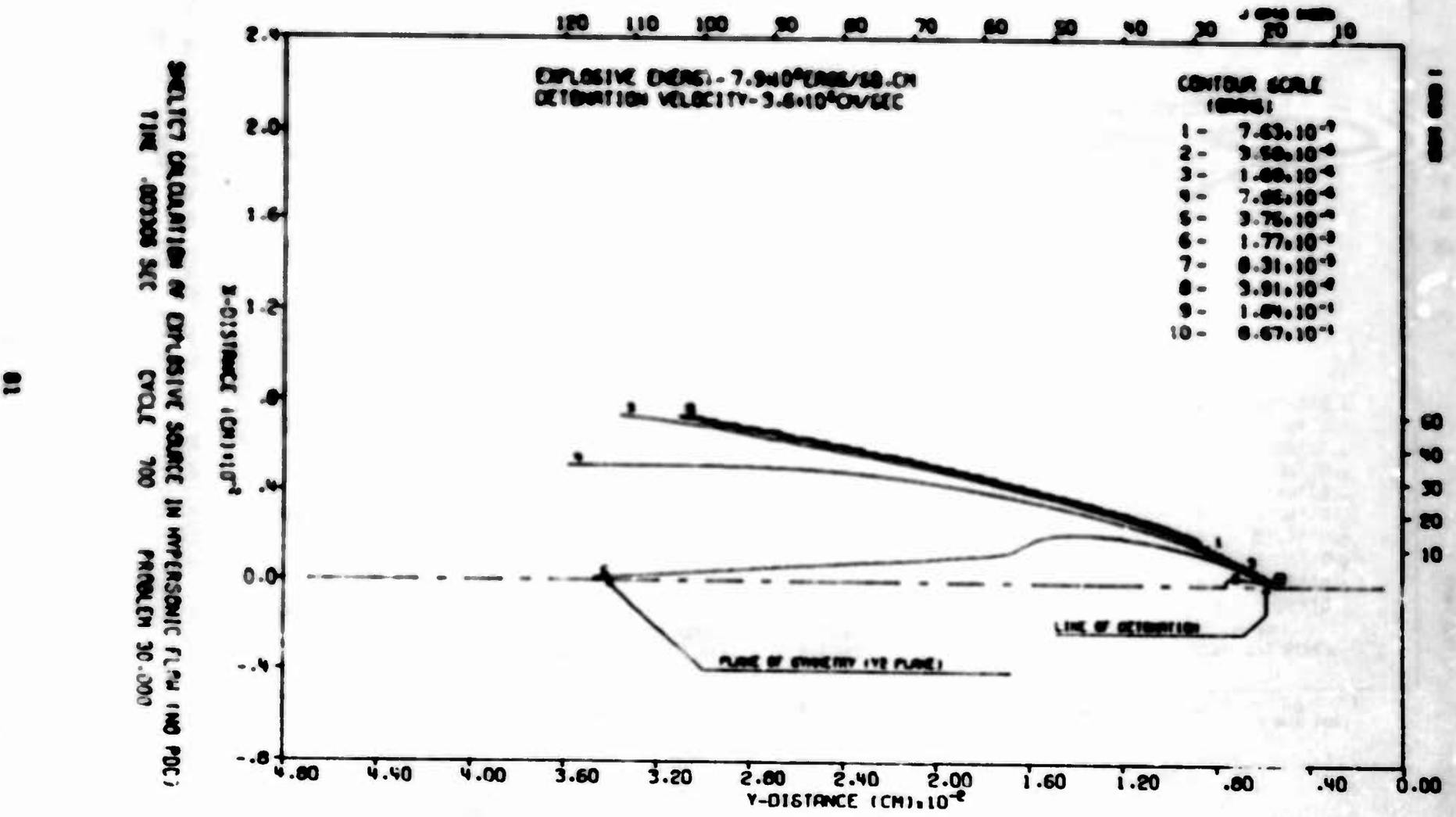


Figure 26. Constant Mass Contours--Species C (Carbon Monoxide)

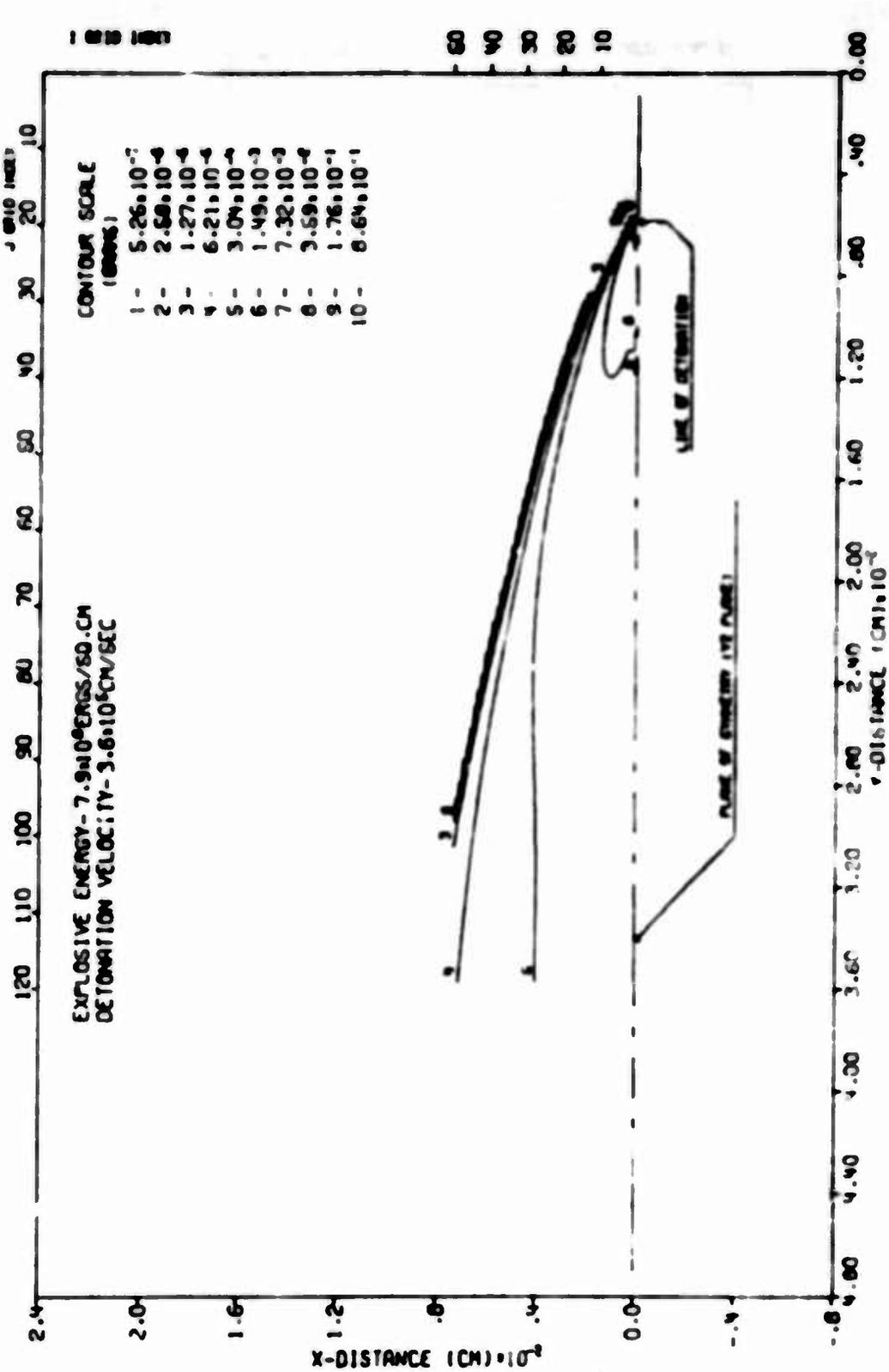


Figure 27. Constant Mass Contours--Species D (Carbon Monoxide)

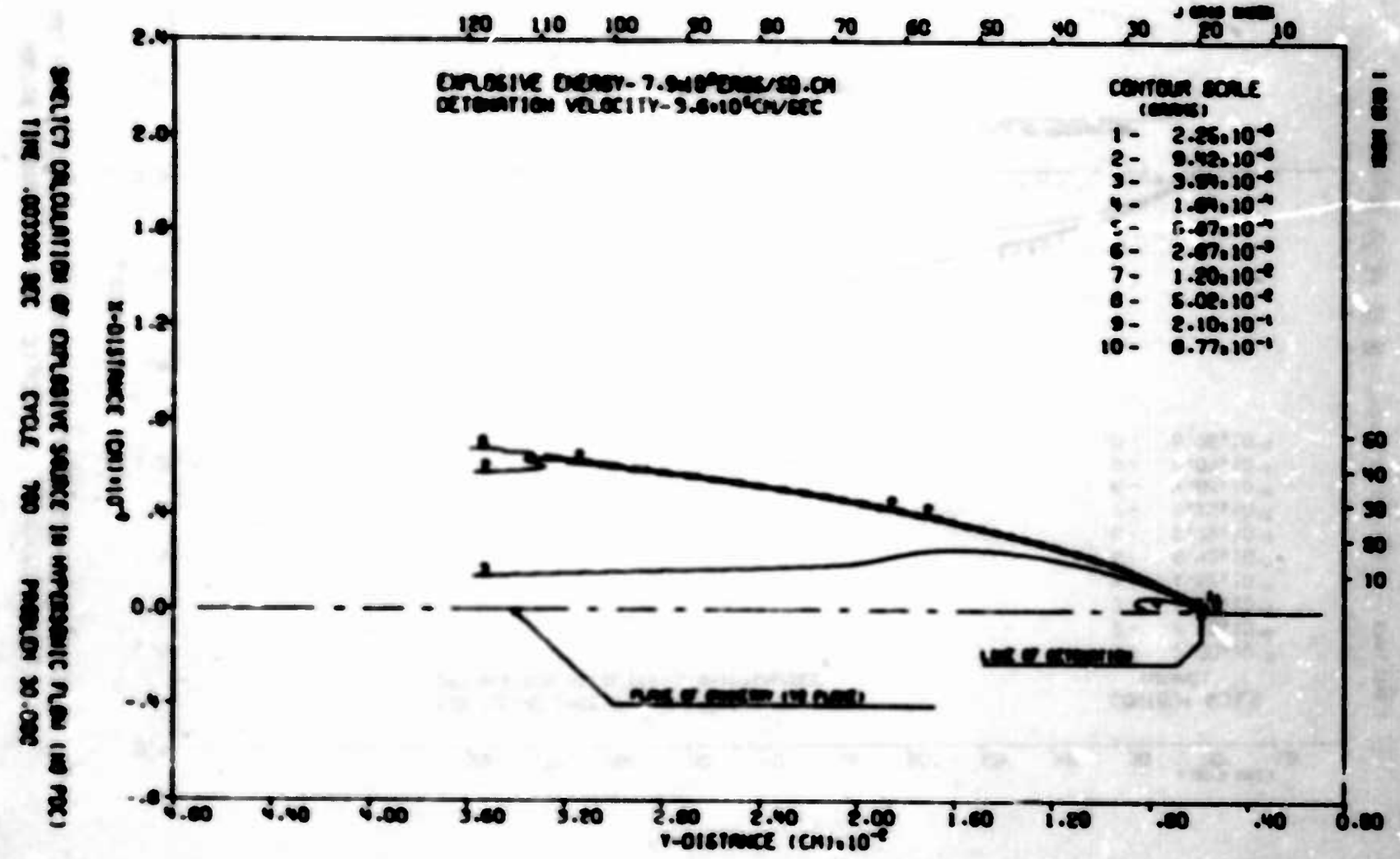


Figure 28. Constant Mass Contours---Species E (Water)

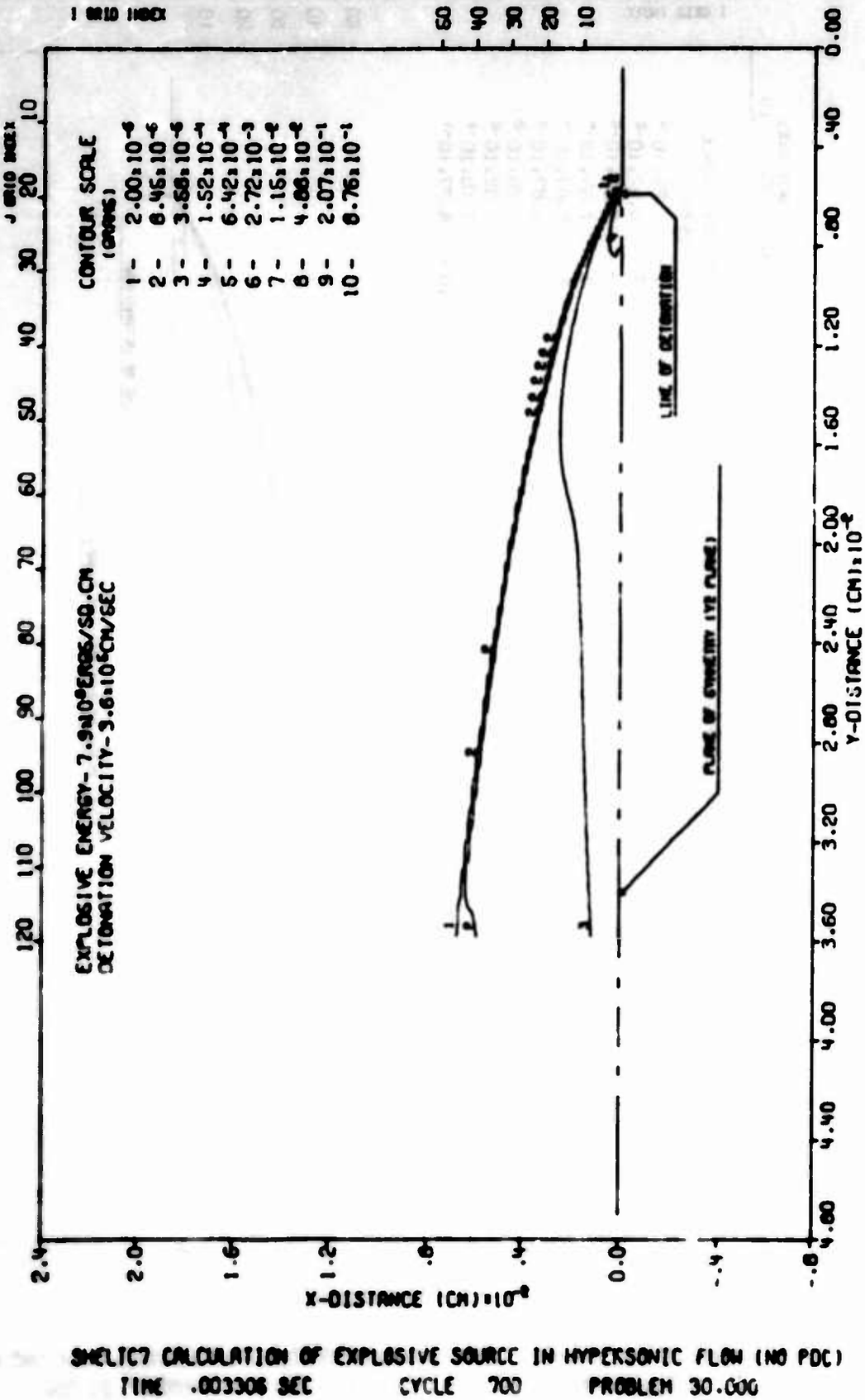


Figure 29. Constant Mass Contours--Species F (Hydrogen)

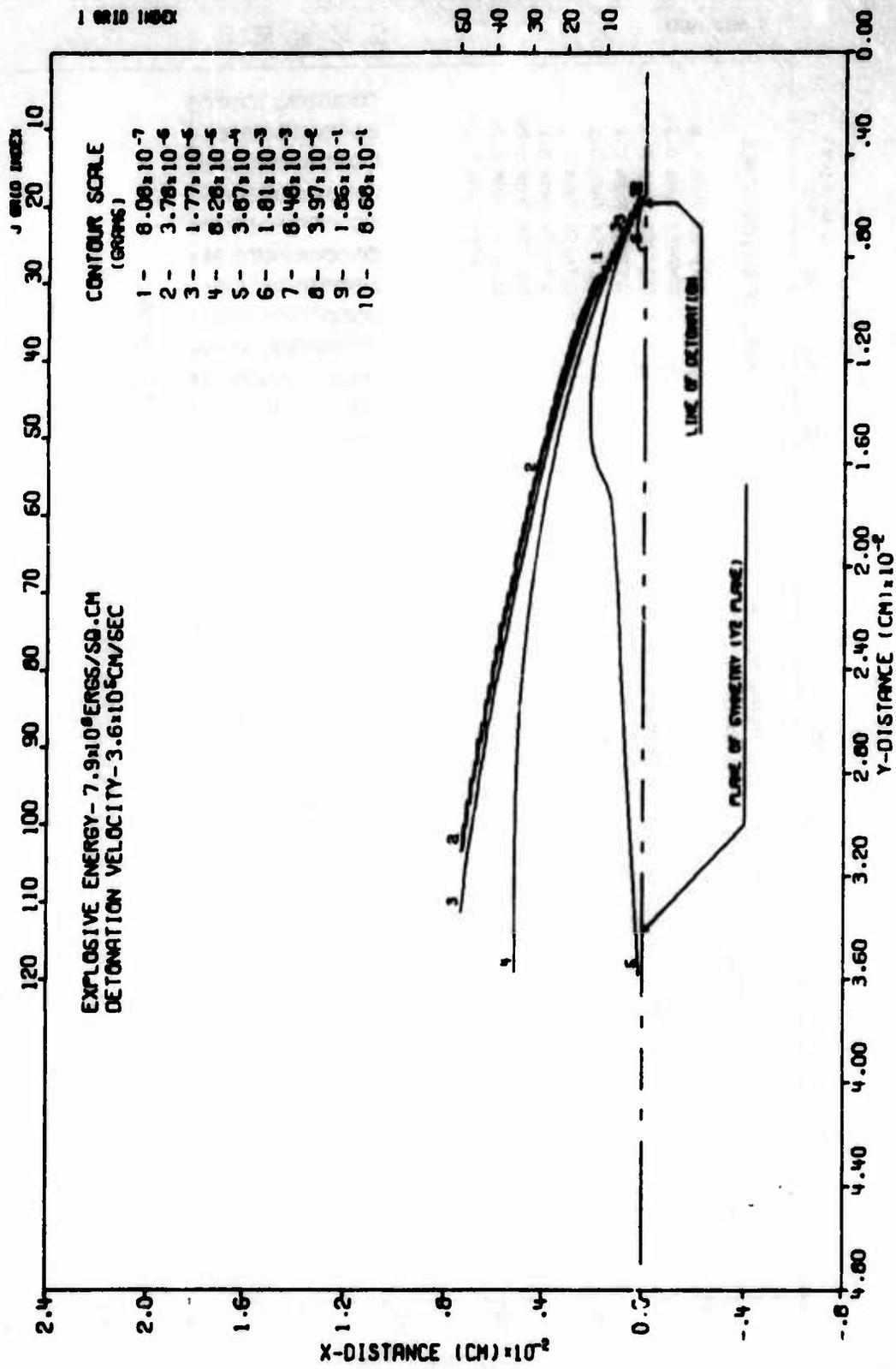


Figure 30. Constant Mass Contours--Species G (Carbon)

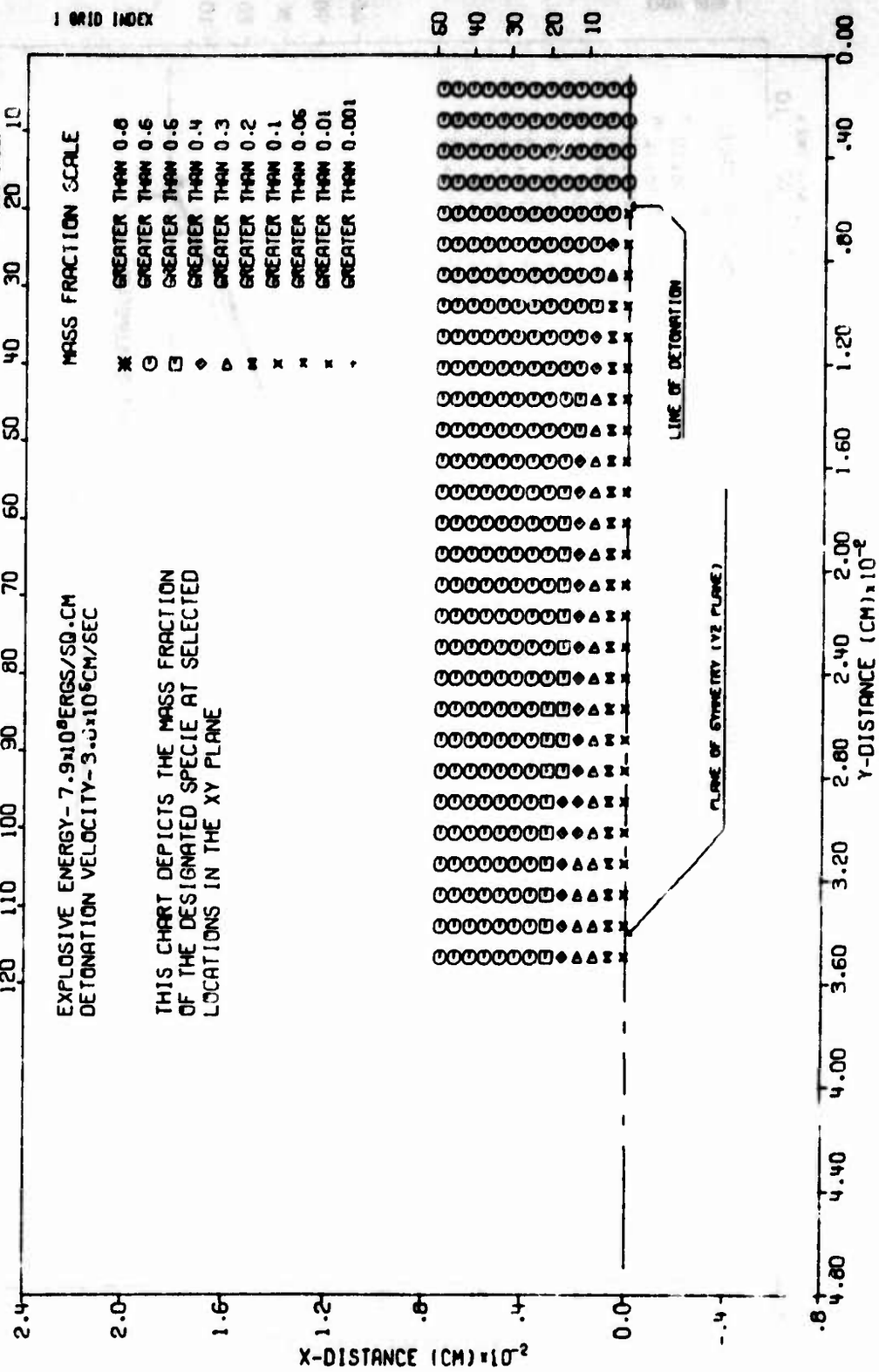


Figure 31. Mass Distribution--Species A (Nitrogen)

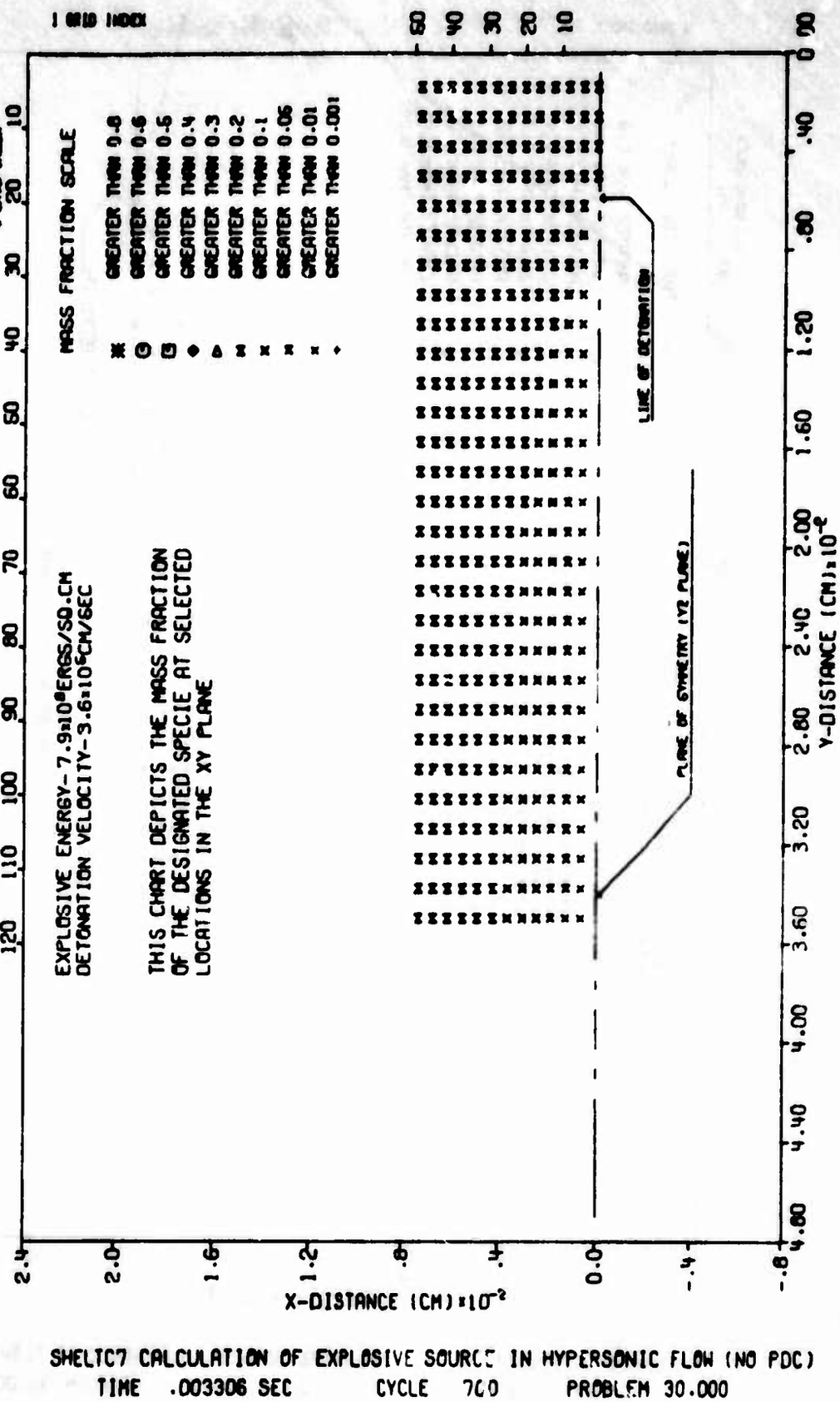


Figure 32. Mass Distribution—Species B (Oxygen)

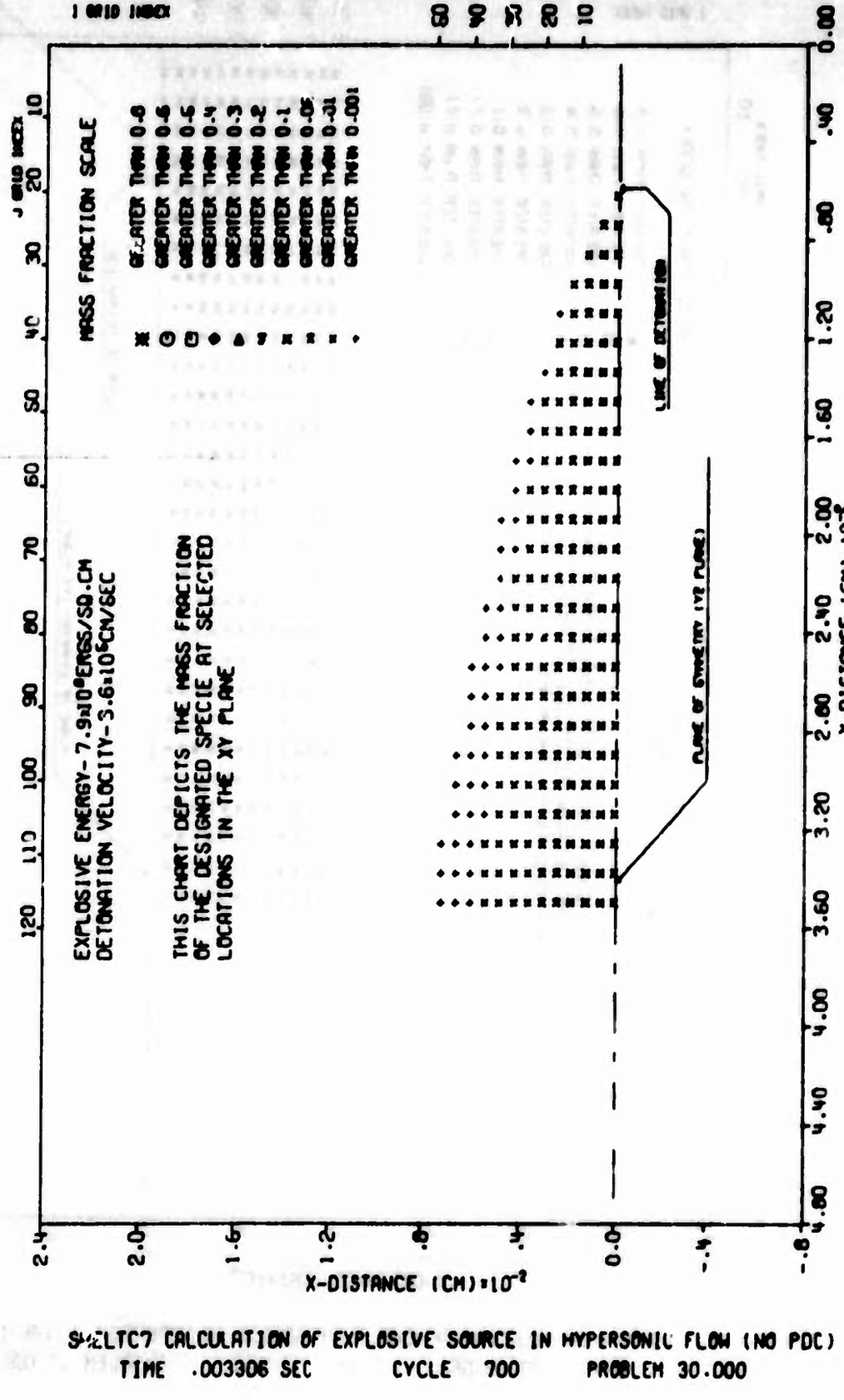


Figure 33. Mass Distribution--Species C (Carbon Monoxide)

SHELTC7 CALCULATION OF EXPLOSIVE SOURCE IN HYPERSONIC FLOW (NO PDC)
TIME .003306 SEC CYCLE 700 PROBLEM 30.000

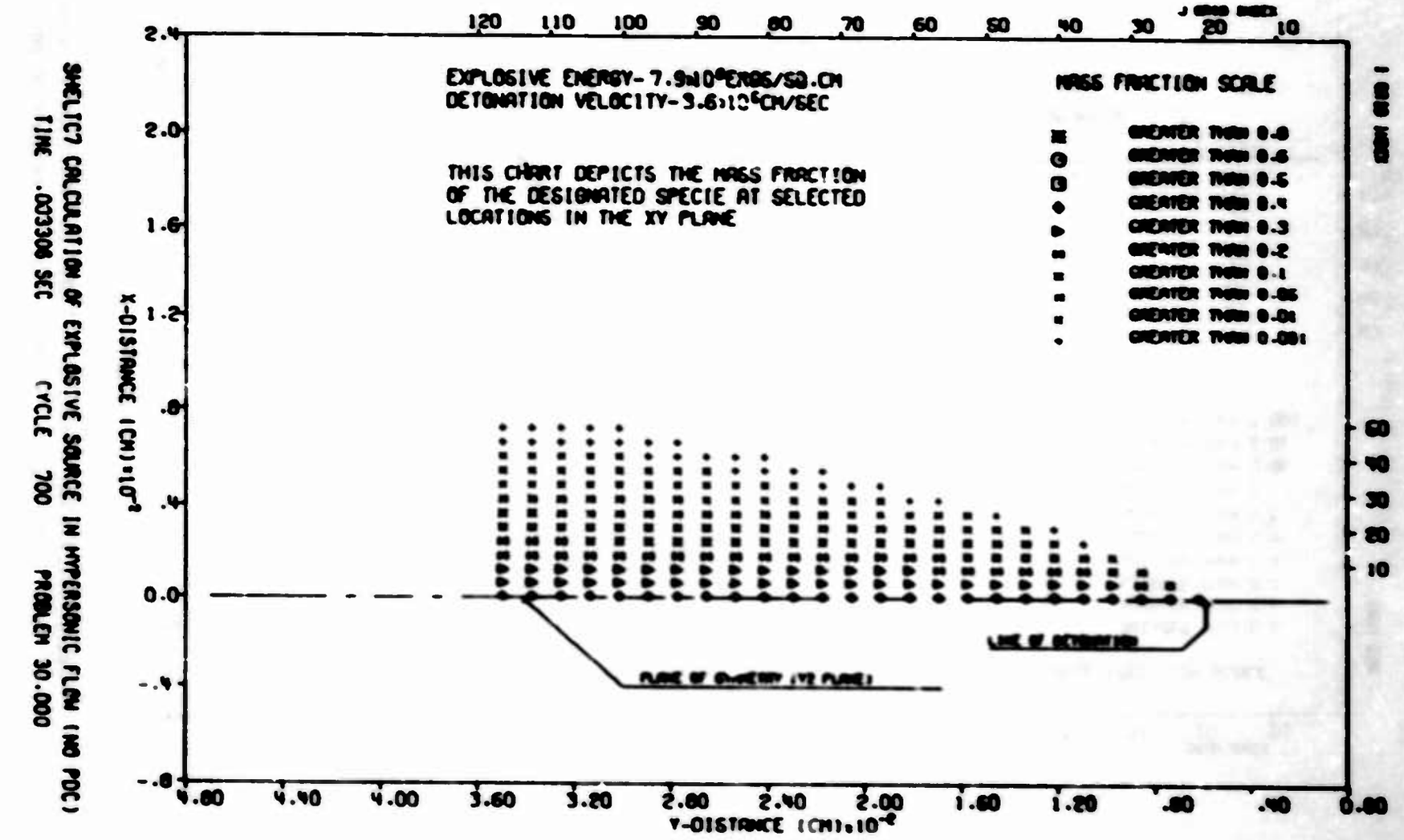


Figure 34. Mass Distribution--Species D (Carbon Dioxide)

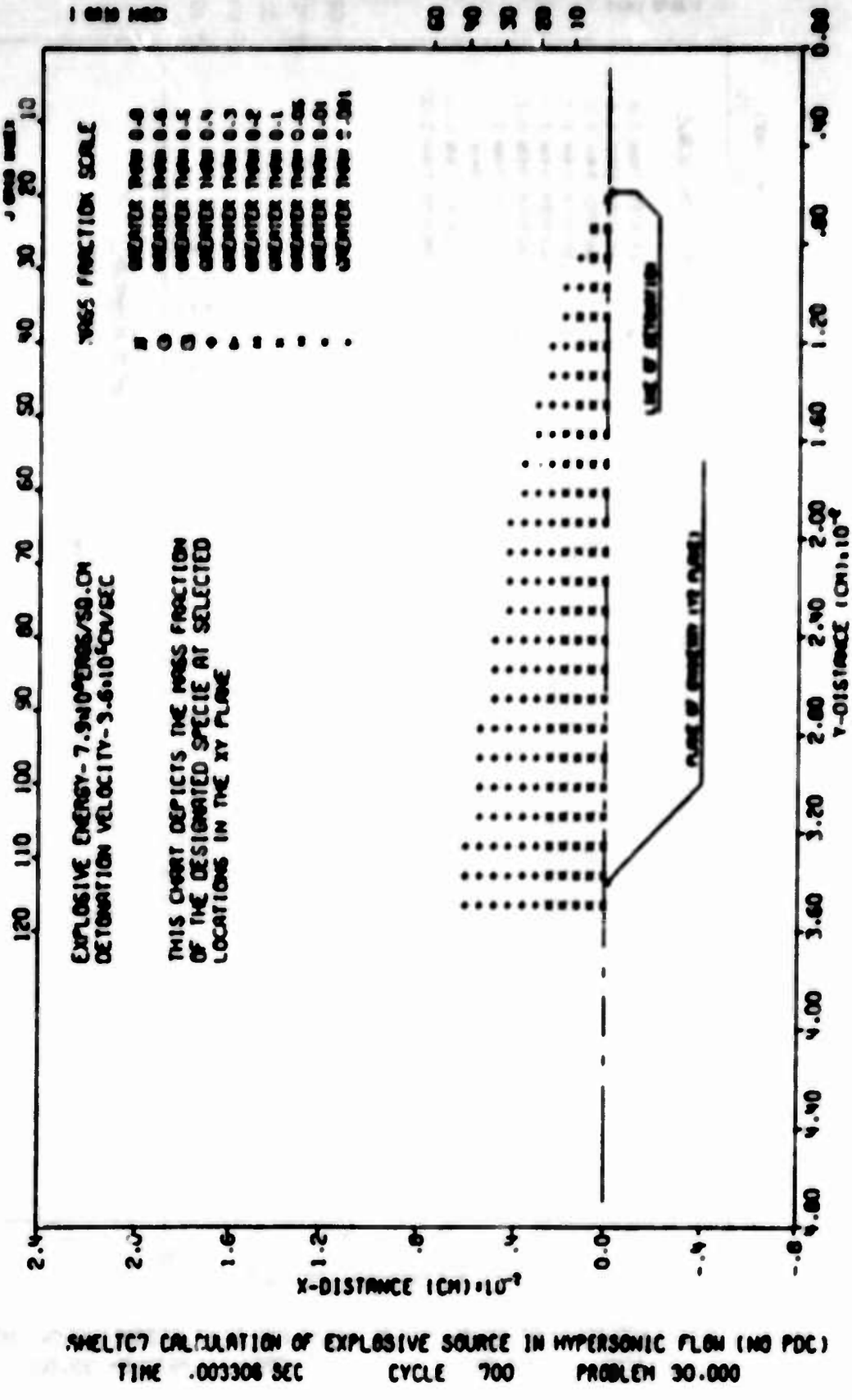


Figure 35. Mass Distribution--Species E (Water)

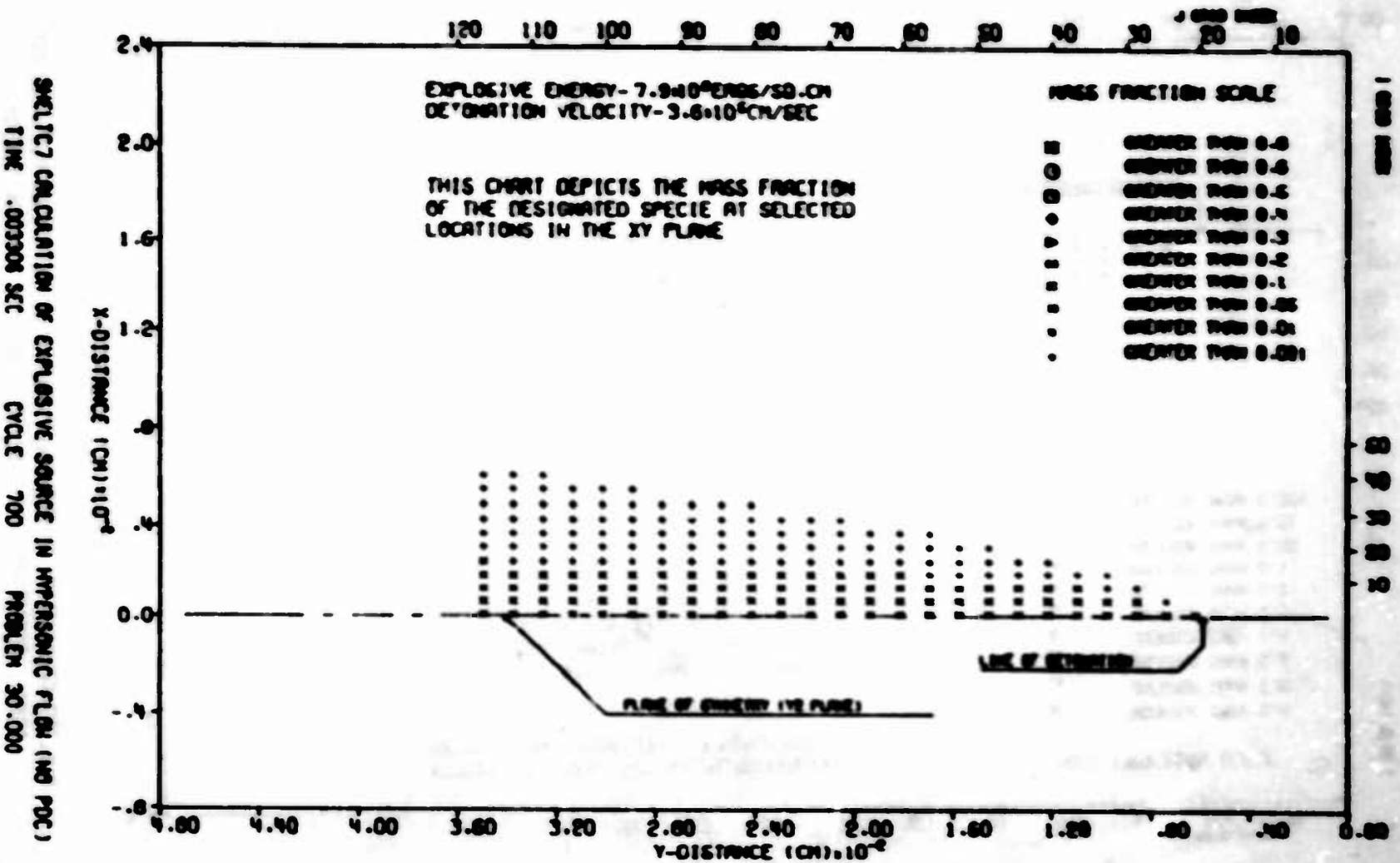
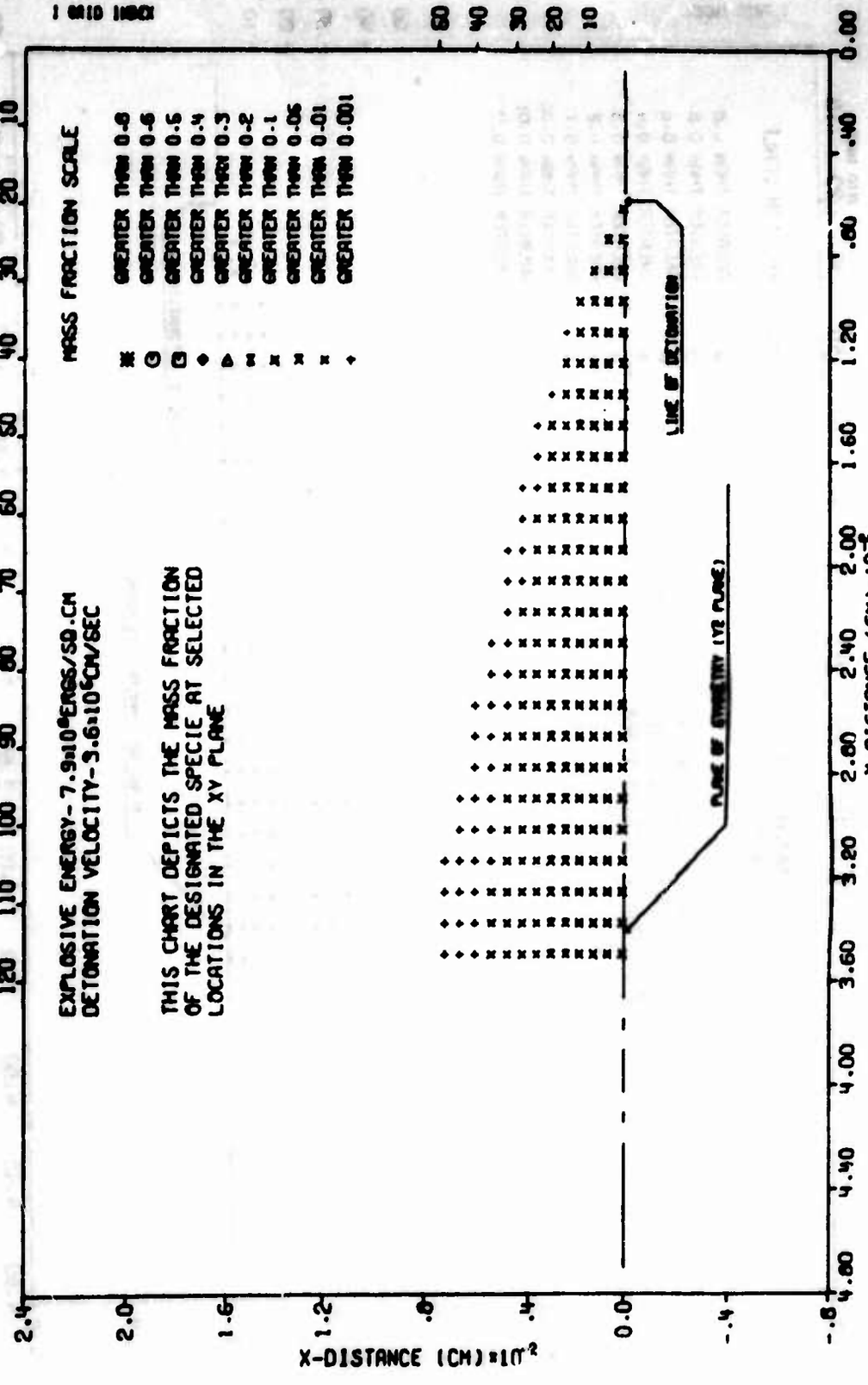


Figure 36. Mass Distribution—Species F (Hydrogen)



SHELTC7 CALCULATION OF EXPLOSIVE SOURCE IN HYPERSONIC FLOW (NO PDC)
TIME .003308 SEC CYCLE 700 PROBLEM 30.000

Figure 37. Mass Distribution--Species G (Carbon)

POST-DETONATION COMBUSTION SCHEME

Several approaches for approximating the PDC were attempted before a successful scheme was finally achieved. First a complete combustion-reaction scheme was tried where all combustion reactions were carried to completion, burning either all the fuel or all the oxygen in a cell once during each time step in an attempt at bounding the problem. Figure 38 shows that the difference between not including PDC and including PDC using the complete combustion scheme is quite drastic in terms of the shock location. However, this resulted in more energy being introduced into the flow from PDC than from the detonation, which is obviously not very representative of the real case. Next, stoichiometric combustion reactions were allowed to proceed to some arbitrarily chosen percent of completion. Here again, to preclude more energy from PDC than from the basic detonation, the precentage had to be set extremely small and there was no way to determine what percentage was correct to use. Next, a crude reaction rate scheme was tried where arbitrary rates were established at which each of the three combustion reactions (C to CO, H_2 to H_2O, and CO to CO_2) were allowed to proceed. These rates were defined as arbitrary functions of the temperature of the cell and the molar concentration of the reactants in the cell. Here again, the arbitrary rate functions led to excessive amounts of energy being deposited during each time step and, in addition, led to the formation of "hot spots." Here the temperature of the cell and the amounts

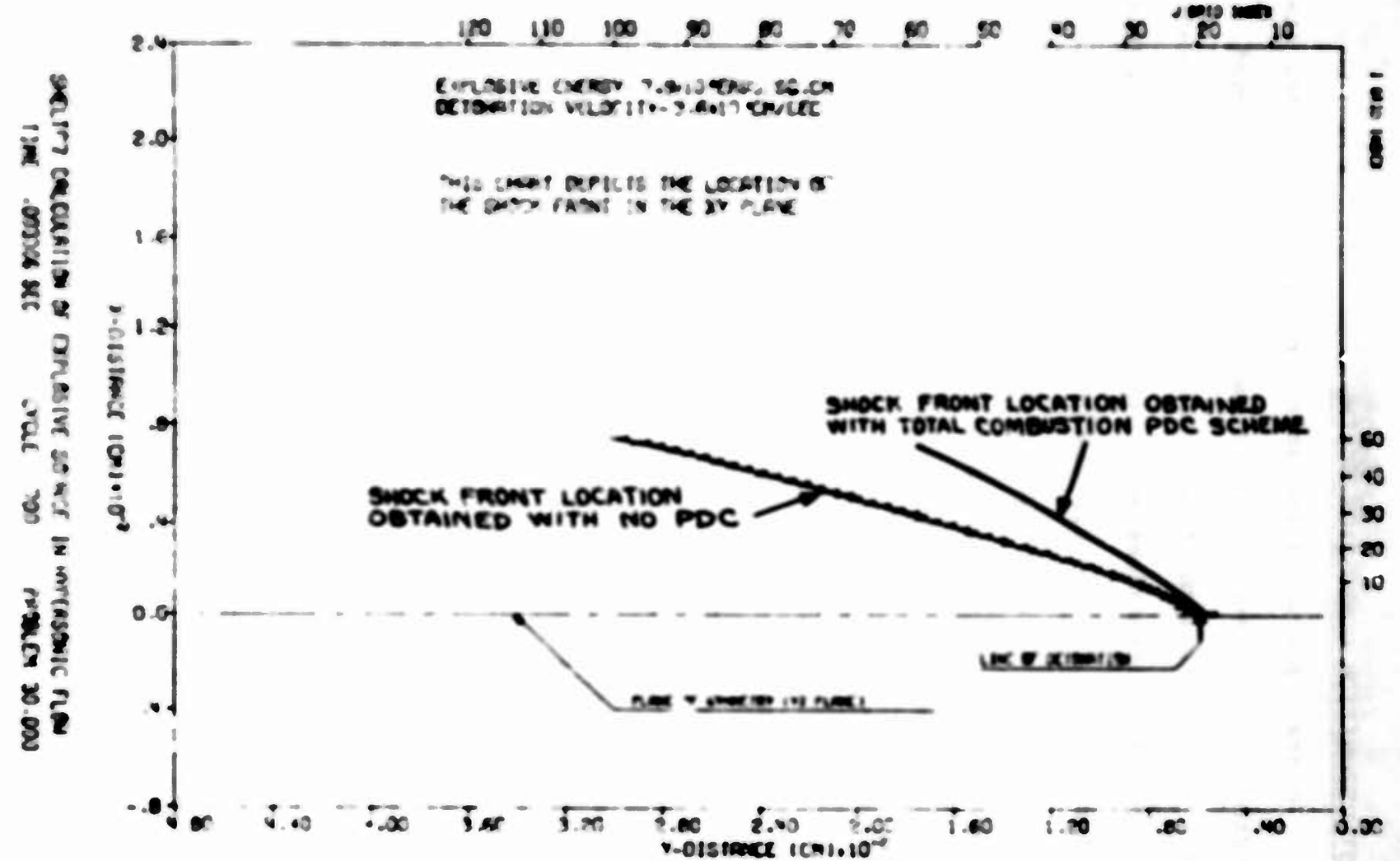


Figure 38. Shock Front Location in the XY Plane

of fuel and oxygen in the cell led to large energy deposition in that cell which, in turn, raised the temperature of the cell. This process continued, causing the cell temperature to grow without bounds. It then became apparent that no satisfactory combustion scheme could be devised without somehow incorporating actual reaction kinetics.

A new approach was taken following a study by Cherry (42). The eight reactions listed in Table VIII were considered significant as a starting point.

TABLE VIII

C, H, O, N Reactions with Third Body

Reaction Number	Reaction	Rate Constant*
1	$CO_2 + M = CO + O + M$	$1 \times 10^{16} \exp[-3500/(RT)]$
2	$H_2O + M = OH + H + M$	$1 \times 10^{19} T^{-1.0}$
3	$CO + M = C + O + M$	$3 \times 10^{16} T^{-0.5}$
4	$H_2 + M = H + H + M$	$7.5 \times 10^{18} T^{-1}$
5	$OH + M = O + H + M$	$2.0 \times 10^{18} T^{-1}$
6	$O_2 + M = O + O + M$	$1.9 \times 10^{16} T^{-0.5}$
7	$N_2 + M = N + N + M$	$1 \times 10^{18} T^{-1}$
8	$NO + M = N + O + M$	$6 \times 10^{16} T^{-0.5}$

*In cm, mole, °K, sec units. Rate is for reverse (exothermic) direction.

No values of the rate constants for the forward (endothermic) reactions were found, so they were approximated by using the Arrhenius rate equation (Ref. 37).

$$R = A\, e^{\frac{-E}{RT}} \tag{55}$$

where

A = frequency factor

E = activation energy (ergs)

T = temperature (°K)

R = ideal gas constant

Considering, now, the relationship between the forward and the reverse reaction rate constants,

$$\frac{R_f}{R_r} = \frac{A_f\, e^{\frac{-E_f}{RT}}}{A_r\, e^{\frac{-E_r}{RT}}} \tag{56}$$

$$R_f = R_r \cdot \left(\frac{A_f}{A_r}\right) \cdot e^{\frac{E_r - E_f}{RT}} \tag{57}$$

If the ratio of frequency factors is assumed to be approximately one and the difference in activation energies is taken to be the combustion energy for the reaction, E_c, then

$$R_f = R_r \cdot e^{\frac{-E_c}{RT}} \tag{58}$$

provides a satisfactory approximation for the forward reaction rate constant. The expressions used to obtain the

forward reaction rate constants for the reactions listed in Table VIII are

$$R_{f1} = R_{r1} \cdot e^{\frac{-6.26 \times 10^4}{T}}$$

$$R_{f2} = R_{r2} \cdot e^{\frac{-5.92 \times 10^4}{T}}$$

$$R_{f3} = R_{r3} \cdot e^{\frac{-1.29 \times 10^5}{T}}$$

$$R_{f4} = R_{r4} \cdot e^{\frac{-5.22 \times 10^4}{T}}$$

$$R_{f5} = R_{r5} \cdot e^{\frac{-5.10 \times 10^4}{T}} \qquad (59)$$

$$R_{f6} = R_{r6} \cdot e^{\frac{-5.92 \times 10^4}{T}}$$

$$R_{f7} = R_{r7} \cdot e^{\frac{-1.13 \times 10^5}{T}}$$

$$R_{f8} = R_{r8} \cdot e^{\frac{-7.54 \times 10^4}{T}}$$

Units of the reaction rate constants are moles/cm^3-sec.

In attempting to approximate a reaction rate controlled combustion scheme the rate controlling reaction must be identified. A set of initial conditions was established (arbitrarily taken from a grid cell in a previous computation) containing initial concentrations of each of the species encountered in the reactions listed in Table VIII and a starting temperature. A simplified Euler Time-Centered Predictor-Corrector numerical integration scheme was used in an auxiliary computation to integrate equation 54 for these 16 reactions using the above initial conditions. The results of this computation indicated that the reactions involving nitrogen (reactions 7 and 8, Table VIII) do not progress significantly over any long period of tine, hence, they can be ignored. The reaction dissociating molecular oxygen to atomic oxygen (reaction 6 forward, Table VIII) was identified as the rate controlling reaction. The reactions consuming atomic oxygen proceeded at a much greater rate than the reaction producing it. Based on this knowledge a new PDC reaction scheme was devised wherein the actual rate of dissociation of O_2 into O is determined from an adjusted cell temperature. This rate, acting over the period of time of the hydrodynamic time step, determines the amount of atomic oxygen which is formed during a time step and is thus available for reacting with the combustible materials in the cell. To determine how much of this available atomic oxygen goes into the combustion of each of the three fuels (C, CO, H_2), a reaction rate for each is computed. The computation of a rate for the C and

CO combustion presents no problem since the molar concentration of each of these in the cell is known. Determining a rate for combustion of H_2 presents a problem because it depends upon the molar concentration of atomic hydrogen in the cell (which is not a stored variable). A way around this difficulty was obtained by assuming that the atomic and molecular hydrogen have come to dissociative equilibrium at the cell temperature which is not an unreasonable assumption since the time constant for attaining this equilibrium is short compared to the hydrodynamic time step for most cell temperatures. With a concentration of atomic hydrogen defined, a rate for the combustion of hydrogen to water can be computed. Thus, quantities of C, CO, and H_2 are burned proportional to the reaction rates computed above for each until all the atomic oxygen is consumed.

A combustion subroutine, called PDC, was constructed to accomplish these reactions in the above manner. The masses of each of the reactants and products in the cell are adjusted the appropriate amount to account for the reactions and the combustion energy produced by the reactions are added to the cell internal energy. This reaction sequence is accomplished in every cell of the grid once during each time step, the subroutine PDC being accomplished immediately after all mass transport in the code has been performed.

This combustion subroutine has been developed using real reaction rates and good approximations for the unknown molar concentrations. It should, therefore, determine the amount

of combustion in each cell quite well. Nevertheless, experimental verification of this combustion approximation would be very helpful.

COMPUTED RESULTS INCLUDING PDC

The mass and energy addition solution obtained in Section III for the energy values of 7.9×10^8 and 7.9×10^9 ergs/cm^2 and detonation velocity of 3.6×10^5 cm/sec was recomputed with the combustion subroutine included. For an explosive sheet with an energy density of 7.9×10^8 ergs/cm^2, the PDC resulted in an approximate total of 4.4×10^3 ergs of energy being deposited in the total grid during a single time step in contrast to approximately 1.4×10^9 ergs being deposited in the grid by the basic detonation during this time. The amount of energy added by the PDC is not sufficient to influence the formation of the blast wave. This fact is borne out by Figure 39 which is a plot of the shock wave shape and location as determined by a SHELTC7 calculation that included the PDC subroutine and is identical with Figure 17, produced by an identical calculation without PDC.

For an energy density of 7.9×10^9 ergs/cm^2 an approximate total of 2×10^7 ergs was deposited into the grid by PDC reactions compared to 1×10^{10} ergs deposited by the detonation during a time step. Although this is a substantial increase in PDC energy deposited throughout the grid over the amount deposited for the 7.9×10^8 energy density, it is still small compared to the detonation energy deposited and the

shock formation is again essentially uninfluenced by the PDC as can be seen in Figures 40 and 41.

Thus, it appears that the post-detonation combustion of the detonation products in air contributes very little to the initial formation of the blast wave. If one were to continue to sum the PDC effect over the entire history of the blast wave, a small effect would perhaps be observed at late times.

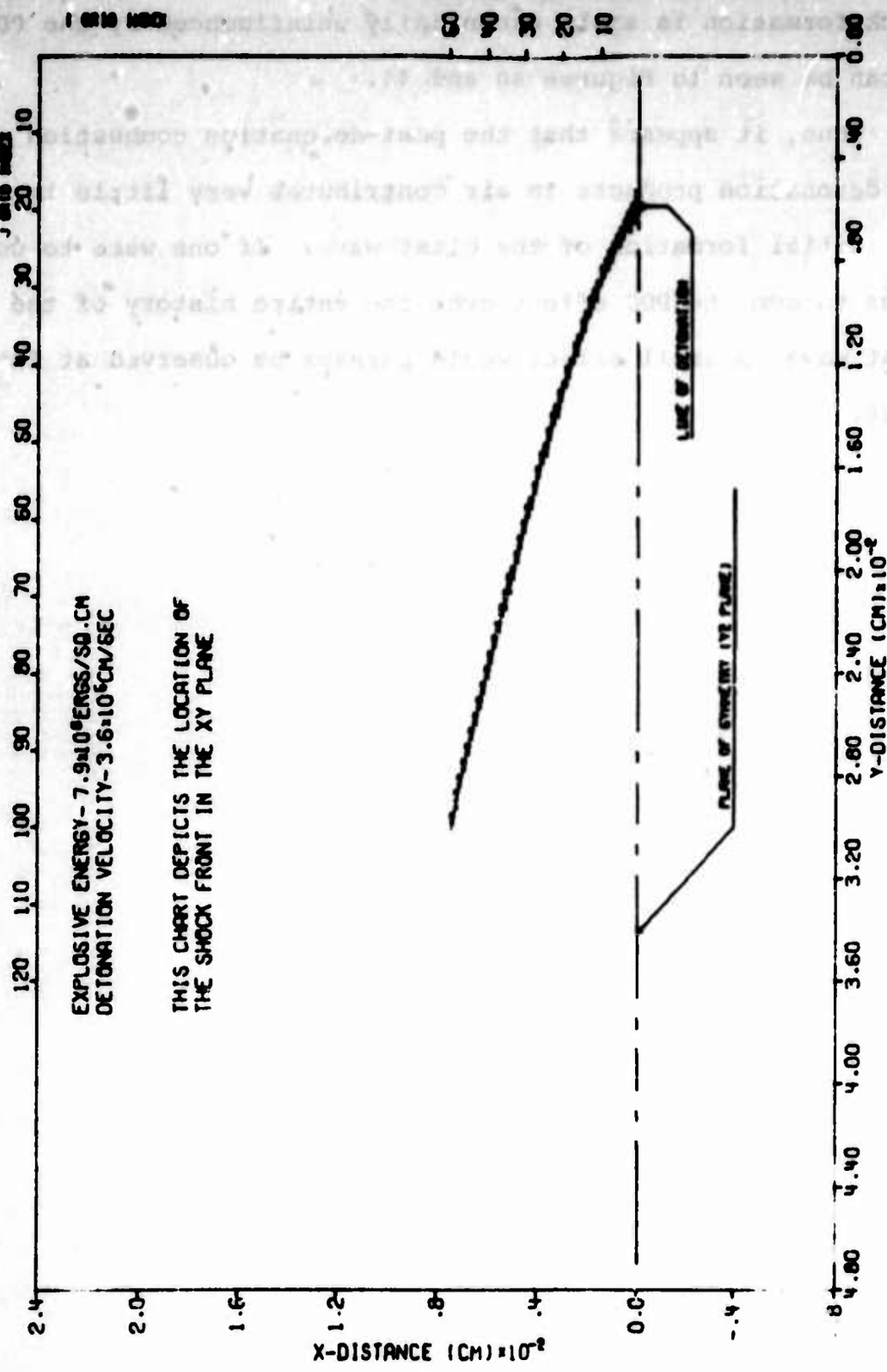


Figure 39. Shock Front Location in the XY Plane

SHELTC7 CALCULATION OF EXPLOSIVE SOURCE IN HYPERSONIC FLOW (WITH PDC)
TIME .005744 SEC CYCLE 1200 PROBLEM 30.000

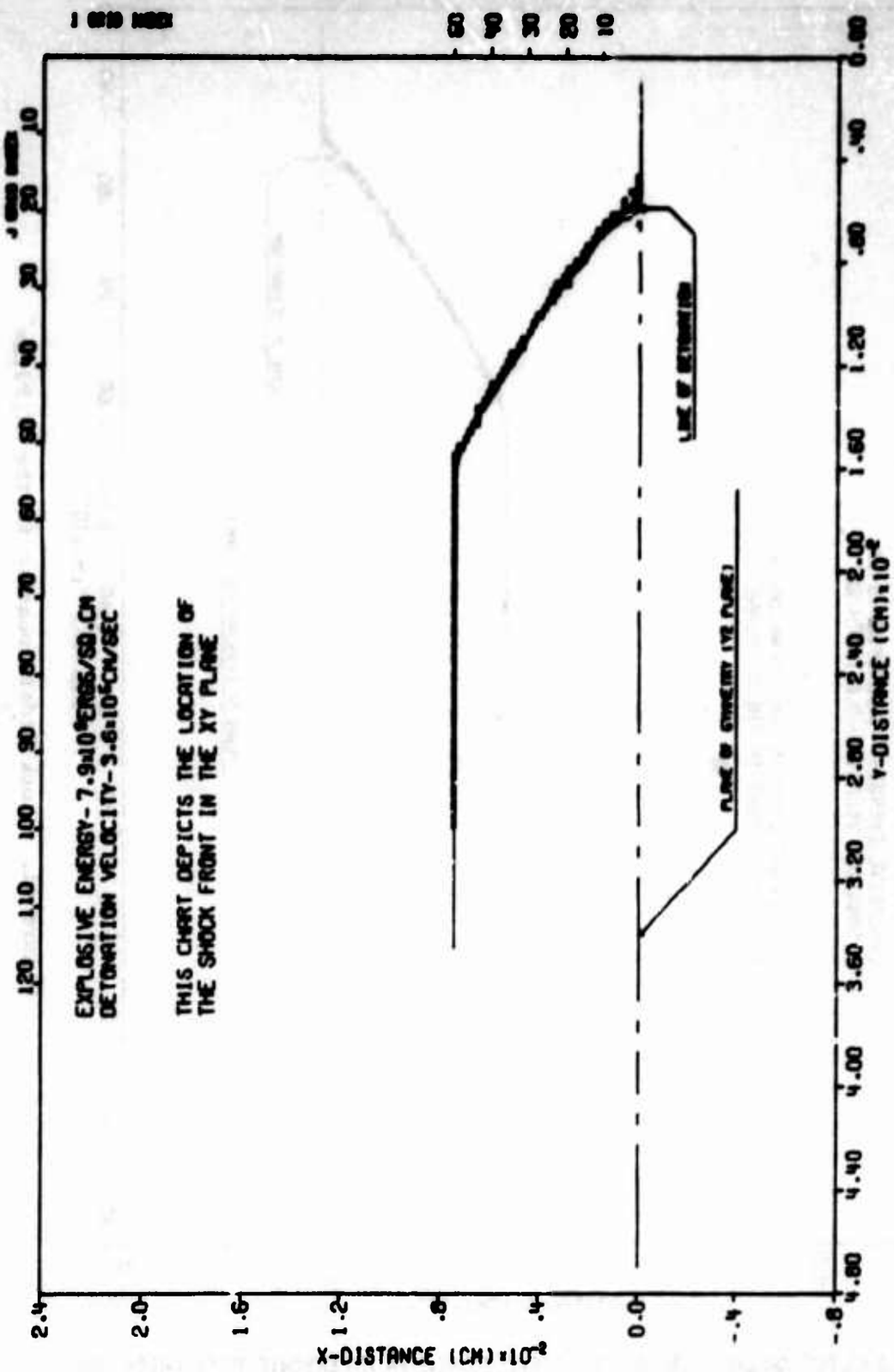


Figure 40. Shock Front Location in the XY Plane

SHELTC7 CALCULATION OF EXPLOSIVE SOURCE IN HYPERSONIC FLOW (NO PDC)
TIME .003296 SEC CYCLE 800 PROBLEM 30.000

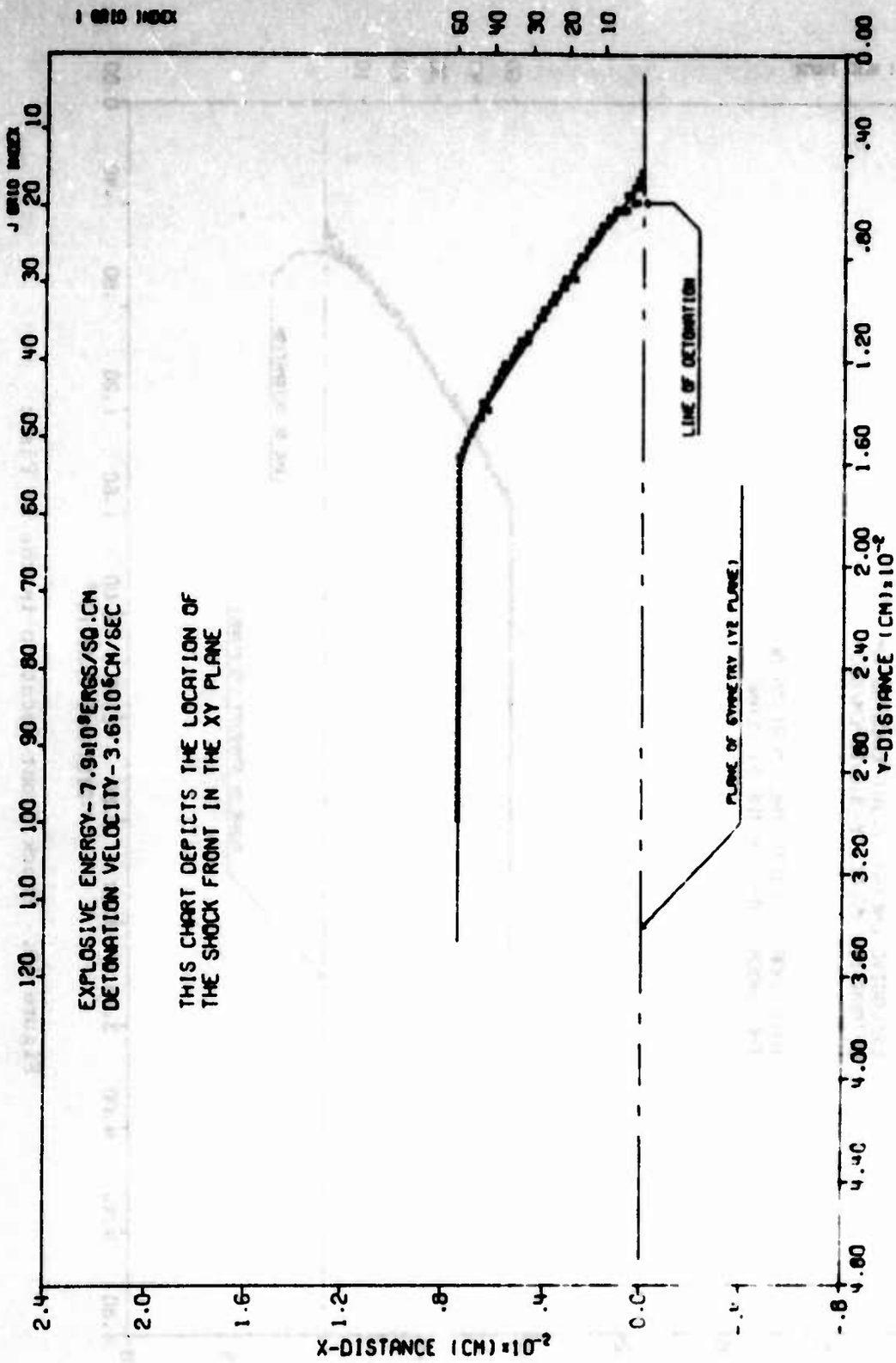


Figure 41. Shock Front Location in the XY Plane

SECTION V

CONCLUSIONS AND RECOMMENDATIONS

The problem of determining the exact shape, location and properties associated with the blast wave from a constant-velocity detonation of an explosive sheet has been found to be extremely complex. It is unlikely that a closed-form analytical solution for even the most idealized form of the problem can be obtained. Numerical computations using hydrodynamic computer codes offer the greatest hope for obtaining a realistic solution.

A solution for the problem, idealized to consider the explosive sheet as releasing only energy and the surrounding medium to be an ideal gas ($\gamma = 1.4$), has been obtained using the SHELLTC computer code. The shape and location of the shock front along with the properties and flow behind the front have been determined for several energies and velocities of detonation.

The shape and location of the shock wave which is formed depend both upon the energy density of the explosive sheet and upon the velocity of detonation. It is clear that the shock forms and remains continually ahead of the line of detonation, the distance in front being determined by the energy density of the explosive sheet and the velocity of detonation. An increase in the energy density causes the shock to move out further in front of the line of detonation while an increase in the velocity of detonation tends to reduce this distance. The shock wave for each energy and velocity

of detonation is near parabolic in shape close to the line of detonation and approaches a fixed angle (the Mach angle) at large distances from the line of detonation. Arguments can be made as to the exact location of the shock front (since in real gases the shock has a finite structure) but these are relatively unimportant so long as a consistent scheme is used for locating the shock in all the computations. However, the spacing between the numerically obtained isobars that form the shock front (see Figure 8) is greater than would be observed in an actual blast wave in air because the "effective viscosity" in SHELLTC spreads the shock over a greater distance than would be observed in the real case.

The pressures behind the blast wave are highest near the line of detonation and are large all along the shock front (see Figure 8). The parameters chosen to determine the "effective viscosity" in SHELLTC (S1, S2, and S3--see Appendix A) tend to show peak pressures slightly lower than an analytical solution for the same problem. However, the values attained here are thought to be accurate within the range of experimental error from any large explosive experiment.

The mass density also is greatest immediately behind the shock where the bulk of the mass behind the blast wave has accumulated. A low pressure, rarefied region forms behind the line of detonation and moves with it as it proceeds along the explosive sheet as can be seen in Figures 42 and 43. Figure 11 shows that in the coordinate system where the

Figure 42. Constant Pressure Contours (Isobars)

SHELLTC CALCULATION OF POINT SOURCE IN HYPERSONIC FLOW
TIME 0.000864 SEC CYCLE 410 PROBLEM 21.000

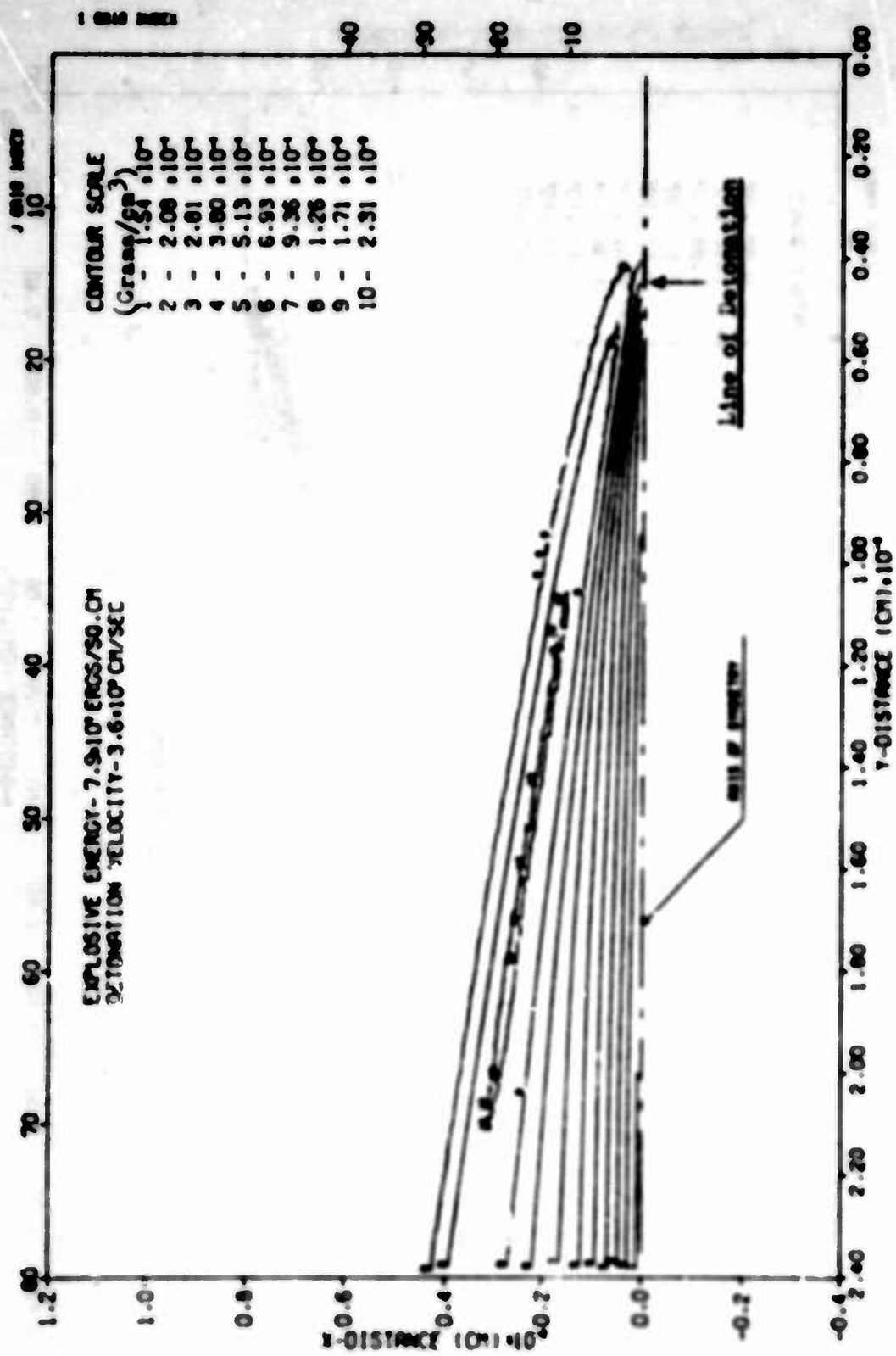


Figure 43. Constant Density Contours

line of detonation is fixed, all flow is essentially from left to right through the grid. However, in the coordinate system attached to the undisturbed gas (line of detonation moves--see Figure 12), the flow of particles in the region behind the shock and in front of the line of detonation (along the explosive sheet) is forward (toward the shock front), but the particle velocity is less than the velocity of detonation. Hence, the detonation overtakes the mass flow and as the detonation approaches, the particle velocity turns outward, away from the explosive sheet. Finally, downstream from the line of detonation, the particle velocity along the explosive sheet is directly away from the line of detonation. Hence, there is a reversal of flow in the region close to the explosive sheet as the detonation passes.

The sheet explosive problem was more realistically modeled by considering the explosive material to release mass as well as energy. A solution for the blast wave from this model was obtained using the multi-material SHELTC7 hydrodynamic code (which was developed for use in this study) along with a temperature-dependent equation of state. The formation of the blast wave was similar to the idealized case, except that the location and curvature of the shock front were altered. In this mass and energy addition solution the shock front forms farther in front of the line of detonation than for the idealized solution but it curves back more sharply producing a shallower shock profile. This solution is probably more representative of the wave from a real explosion.

A scheme for estimating the contribution to the blast wave of the post-detonation combustion was devised based on an "effective mixing" which is inherent in the SHELTC7 code. A method for approximating the combustion reactions and determining the energy released by these approximate reactions was developed and used to determine the effect of the PDC on the blast wave. The results of this work indicate that post-detonation combustion has little or no effect on the formation of the blast wave. Even though significant amounts of energy may be released by the combustion, the amounts are small compared to the detonation energy released and thus contribute little to the formation of the wave. Even though there is some doubt about the validity of the mixing used in determining the amount of PDC, the above conclusion remains unchanged since a substantial increase in PDC energy would be required before any noticeable change in the blast wave would be obtained. Since the combustion approximation scheme is based on real reaction rates or reasonable approximations thereof, it can be accepted as reasonably valid. Thus, although some error is introduced by the combustion approximation, the major source of uncertainty in determining the PDC energy released is the mixing of mass species which determines the amount of materials available for PDC. Substantial variation in the mixing can occur with only a minimum effect on the formation of the blast wave.

An expression was empirically obtained (Equation 23) which approximates the shape and location of the shock front

as determined by the idealized solution for a given energy density and velocity of detonation. If approximate information on the shock front location is all that is required, then this expression can be used to obtain it directly. A better estimate of the actual shock location can be obtained by adjusting downward the shock profile obtained from Equation 23 an amount comparable to that observed in Figure 22. If additional information about the properties of a particular blast wave is required (such as pressures, particle velocities, etc.), a computer solution following the methods described here is necessary. In view of the negligible contribution of the PDC it need not be included in the computations.

SUGGESTIONS FOR ADDITIONAL RESEARCH

The work accomplished during this study is of sufficient importance and has wide enough application to warrant additional study and expenditure of research effort. Suggestions for further work are described in the following paragraphs.

The empirically obtained expression describing the location and shape of the shock wave (equation 23, Section II) has been developed from limited data. The suitability of this expression for explosive energy densities beyond 7.9×10^9 ergs/cm^2 has not been verified. In addition, it appears that a similarity in shock-wave profiles exists for different energy densities and for different velocities of detonation. It may be possible to obtain a single nondimensional

expression (nondimensionalized to energy and detonation velocity) which would correctly describe the entire family of shock profiles in the xy plane. An effort to develop such a nondimensional expression or at least to develop an empirical expression for predicting the shock front shape and location from a larger data base is warranted. Some additional computations to obtain a larger data base may be required.

A detailed study and definition of the mass transport, mixing, and diffusion properties of SHELTC7 (or other multimaterial Eulerian hydrodynamic codes) would be a significant contribution. Functional relationships should be developed which would describe the mixing of mass species as functions of hydrodynamic flow, species concentrations, grid cell size, etc. Correlation of these relationships with a well designed experiment would be particularly valuable. A meaningful experiment suitable to this goal can be accomplished in a supersonic wind tunnel where a thin flat plate with ports along its leading edge could be used to inject a second mass specie directly in opposition (along the same centerline) to the incoming supersonic flow of the first mass specie. Boundary layer effects would be minimized in this experiment and a valid calculation of the hydrodynamic flow can be accomplished. Sampling of the concentration of the two species of mass downstream from the point of injection will provide experimental data with which to correlate the mixing predicted by the code. Modifications to the code which would improve its capability to model real-flow mixing would be a valuable contribution.

Some studies of the detonation of explosive materials in confined volumes have been made (25). Additional experiments more carefully designed for correlation of calculations should be performed. A series of cylindrical charges of TNT should be detonated in a large cylindrical detonation chamber and these events calculated both in an idealized manner using SHELLTC and then more realistically using SHELTC7. The contribution of PDC can be experimentally determined by purging the chamber with nitrogen prior to one of the detonations. A later experiment in which the chamber is purged with oxygen will provide additional valuable information. Calculation of these experiments with SHELTC7 will determine the capability of the PDC subroutine to predict the contribution of PDC to the blast wave.

Finally, detonation of sheets of explosives (preferably TNT) in open air would provide the most valuable data against which to test the calculational ability of these codes. Here again, the contribution of PDC in the formation of the blast wave can be examined experimentally by enclosing the explosive sheet in a bag made of thin transparent plastic purged with nitrogen or oxygen. Measurement of pressures at selected points corresponding to locations in the computational grid would be required. High-speed color photography of the traveling blast wave would provide data for direct correlation of shock front shape and location. It would be hoped that in a carefully planned experiment some indication of the

location and extent at which PDC occurs could be detected in these color photographs.

APPENDIX A

THE SHELLTC CODE

The SHELLTC code is a two-coordinate system expansion of SHELL OIL which is the basic code of the SHELL family of hydrodynamic codes. The SHELL family of codes evolved directly from the particle-in-cell (PIC) method of computation developed by F. H. Harlow (15) at Los Alamos in 1955. Development of the SHELL computational scheme, formulation of this scheme into a working code, and conversion from particle to continuum are credited to B. E. Freeman, W. E. Johnson and J. M. Walsh. The most detailed description of the SHELL codes is provided by Whitaker, Nawroki and Needham (16). The following discussion of SHELL follows, in part, that document.

The equations solved in SHELL are the 2-D axisymmetric partial differential equations for nonviscous, nonconducting compressible fluid flow. These equations, including the equation of state, are

Mass

$$\left(\frac{\partial}{\partial t} + \vec{u}\cdot\nabla\right)\rho + \rho\nabla\cdot\vec{u} = 0 \qquad \text{(A1)}$$

Momentum

$$\rho\left(\frac{\partial}{\partial t} + \vec{u}\cdot\nabla\right)\vec{u} + \nabla p + \rho\nabla\phi = 0 \qquad \text{(A2)}$$

Energy

$$\rho\left(\frac{\partial}{\partial t} + \vec{u}\cdot\nabla\right)E + \nabla\cdot p\vec{u} + \rho u\cdot\nabla\phi = 0 \qquad \text{(A3)}$$

Equation of State

$$p = p(\rho, I) \qquad (A4)$$

where

ρ = density in gms/cm^3

$\vec{u}$ = velocity in cm/sec

p = pressure in dynes/cm^3

E = specific total energy in ergs/gm

$= I + 1/2\ u^2$

I = specific internal energy in ergs/gm

t = time in sec

ϕ = potential of external force field in ergs/gm

SHELL-OIL is a one material, pure Eulerian code. It solves the hydrodynamic equations by dividing the region occupied by the fluid into a mesh of fixed cells. The fluid is then described at any instant of time by specifying the velocity, density, and internal energy for each cell. These values are considered to be known at the center of each cell and are constant over it. This forced homogeneity of the cell introduces a false diffusion which poses no problem in this one material code but becomes a major disadvantage in multi-material versions of SHELL.

Other disadvantages of SHELL-OIL are the general inability of Eulerian codes to resolve fine detail moving with the fluid, since the mesh is fixed in space, and the inability to follow material interfaces. However, Eulerian codes enjoy

the major advantage of being able to solve the hydrodynamic equations even in the presence of large fluid distortions.

To begin a SHELL-OIL calculation, the problem must first be generated by CLAM, the generator code for SHELL. CLAM sets up the mesh, giving each cell its dimensions and the following quantities: a radial and an axial velocity component, a mass, and a specific internal energy.

The cell quantities are derived from data entered in groups of packages. These data include the type of material, dimensions, velocity components, density, and specific internal energy of the package. To describe as many geometries and energy, density and velocity distributions as possible, CLAM places N^2 particles into each cell, where $1 \leq N \leq 20$ and is also included in the package data. Each cell is divided into N^2 equal parts, and a particle is placed at the center of each area. Taking this package data, a fit assigns each particle two velocity components--a density and a specific internal energy. Then the mass of each particle is computed: the density times the volume of that part of the cell containing the particle. The mass of the cell is the sum of the masses of all the particles in the cell. Both momentum components are calculated as the sum of the individual momentum components of each particle in the cell. The internal energy of each cell is the sum of the internal energies of all the particles in the cell. Finally, these cell quantities are converted to two velocity components and specific internal energy.

There are five geometries available in CLAM such that mass may be loaded into the grid in zones with the following shapes: rectangle, triangle, ellipse, perturbed ellipse and circle. These zones are in the r-z plane and take their 3-dimensional counterparts upon rotation about the z-axis.

After all packages have been processed, CLAM puts all cell dimensions and quantities on tape.

This output provides the starting conditions for the SHELL calculation. The SHELL calculation advances explicitly in steps or cycles. That is, the cell quantities are calculated for time $t + \Delta t$ in terms of those at time t. The solution of the hydrodynamic equations is divided into two phases. In the first, the Eulerian field functions are updated, considering pressure effects alone while neglecting material transport due to fluid motion. In the second phase, continuous mass flow across cell boundaries, new cell densities, velocities, and specific internal energies are calculated. Mass, momentum, and total energy are conserved.

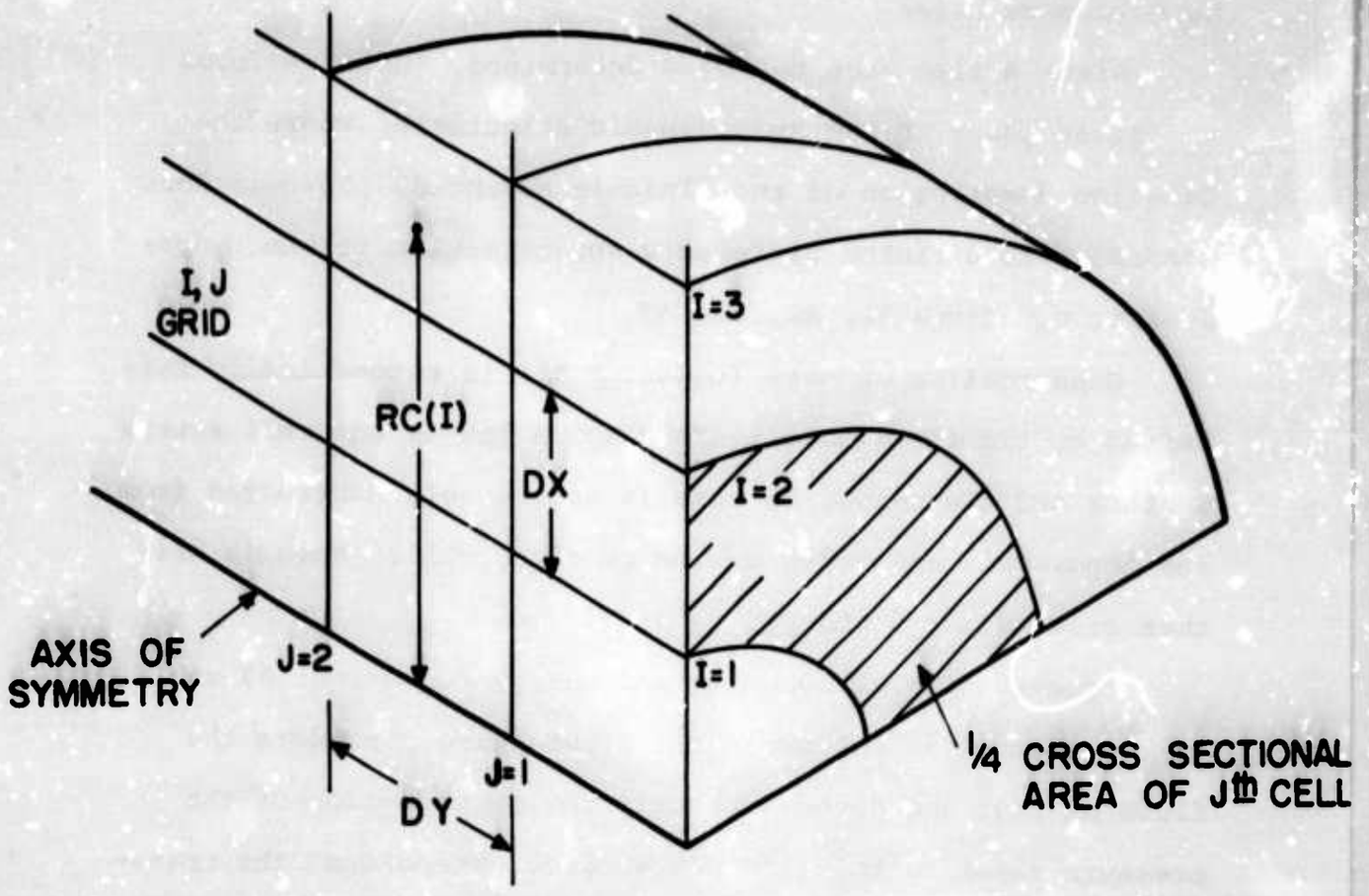
There are three time-control conditions that govern the time step, Δt. They are

a. The Courant condition

b. The maximum of $|\frac{u}{\Delta r}|$ and $|\frac{v}{\Delta z}| < \frac{1}{\Delta t}$

c. A negative time-step option

The first two conditions prohibit the transmission of a signal or mass across more than one cell in one time interval. This requirement is imposed by considerations of stability



RC(J) is radial distance to center of a cell in the Ith column.

Axial dimension of each cell is DY.

Radial dimension of each cell is DX.

Cross-sectional area of each cell is TAU(I) where

$$TAU(I) = 2 \cdot \pi \cdot RC(I) \cdot DX$$

Volume of each cell is

$$VOL(I) = 2 \cdot \pi \cdot RC(I) \cdot DX \cdot DY$$

Figure A1. SHELL Zoning--Two-Dimensional Axisymmetric Case

of the finite difference equations. The third condition will be discussed below.

After a time step has been determined, SHELL performs the first phase of the hydrodynamic calculation where the Eulerian description of the fluid is advanced. This is done according to a finite difference approximation to the hydrodynamic equations A1, A2, and A3.

Conservation of mass (equation A1) is automatically satisfied by the fluid model. Mass which leaves one cell enters another and the change in mass is accordingly subtracted from the donor cell and added to the receiver cell. Mass is neither created nor destroyed.

Conservation of momentum and energy (equations A2 and A3) is treated as follows: the first phase considers the fluid at rest and determines only the contribution of the pressure terms to the time derivative. Therefore, the transport term (second term on the left-hand side) of each equation is dropped. Since the hydrodynamic quantities are defined at the cell centers, the equations in finite difference form become

Momentum

$$\tilde{u}_k = u_k^{(n)} + \frac{1}{\rho_k^{(n)}} \frac{PL^{(n)} - PR^{(n)}}{\Delta r} \Delta t \qquad \text{(A5)}$$

$$\tilde{v}_k = v_k^{(n)} + \frac{1}{\rho_k^{(n)}} \frac{PB^{(n)} - PA^{(n)}}{\Delta z} \Delta t \qquad \text{(A6)}$$

Energy

$$\tilde{I}_k = I_k^{(n)} + \frac{p_k^{(n)}}{\rho_k^{(n)}}\left[\frac{VB^{(n)} - VA^{(n)}}{\Delta z} + \frac{VB - \tilde{V}A}{\Delta z}\right.$$

$$\left. + \frac{2}{r_i + r_{i-1}} \frac{UL^{(n)} - UR^{(n)}}{\Delta r} + \frac{\tilde{U}L - \tilde{U}R}{\Delta r}\right]\frac{\Delta t}{2} \qquad \text{(A7)}$$

where

$$PL = \frac{p_k + p_{kL}}{2} \qquad PR = \frac{p_k + p_{kR}}{2}$$

$$PB = \frac{p_k + p_{kB}}{2} \qquad PA = \frac{p_k + p_{kA}}{2}$$

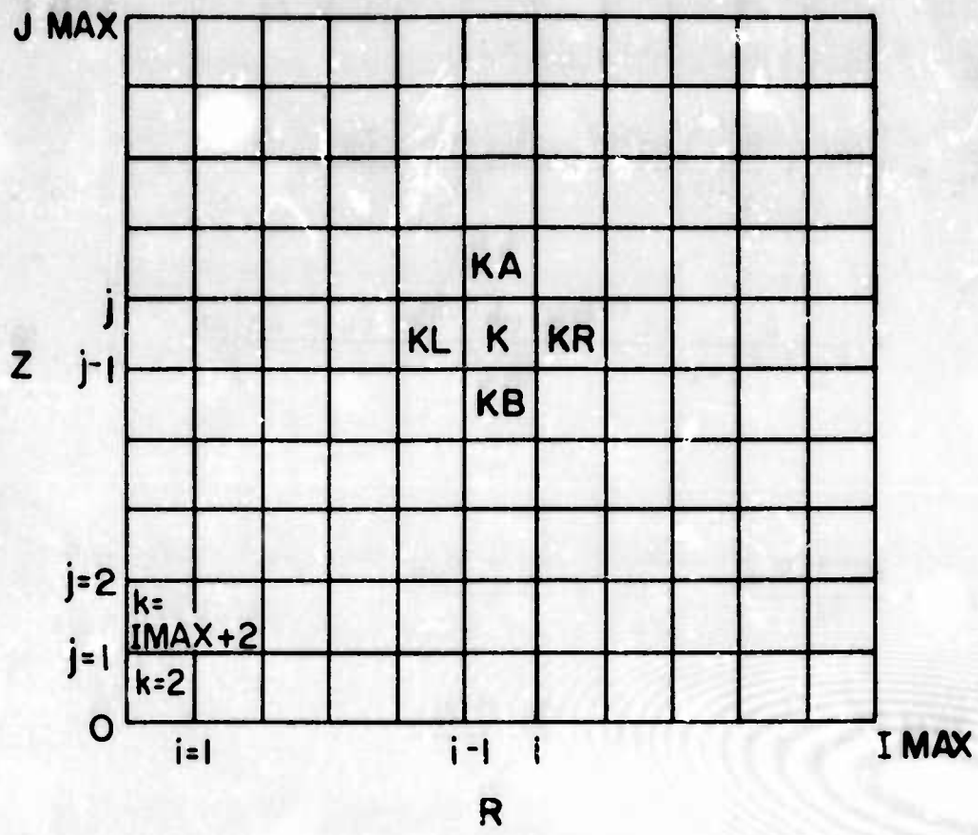
$$VB = \frac{v_k + v_{kB}}{2} \qquad VA = \frac{v_k + v_{kA}}{2}$$

$$UL = \frac{u_k(r_i + r_{i-1}) + u_{kL}(r_{i-1} + r_{i-2})}{4}$$

$$UR = \frac{u_k(r_i + r_{i-1}) + u_{kR}(r_{i+1} + r_i)}{4}$$

and k is the index of the cell center and i is the index of the right-hand boundary of cell k (see Figure A2).

Two passes in succession are made through the first phase of calculation of each cycle. In the first pass, tentative new values of velocity are calculated for a full time step. The internal energy is updated by half a time step, using the old velocities. In the second pass, the internal energy is updated another half-time step, using the tentative



i is the right boundary of cell k

j is the top boundary of cell k

kA is the cell above cell k

etc.

Figure A2. The SHELL Mesh

new velocities calculated in the first pass. The energy equation is treated in this fashion from considerations of stability as well as greatest accuracy in the behavior of fluid entropy. The energy of the system is conserved identically, even in the finite difference approach.

In both passes, the internal energy is checked to see if the updated value has become negative. If it has, an option exists to integrate back to the original configuration of the

fluid at the beginning of the cycle by using the negative Δt. Then the integration goes forward again with a smaller Δt which has been calculated and which will keep the internal energy positive.

In the second phase of the hydrodynamic calculation, continuous mass transport is calculated according to the equation for the conservation of mass, which in finite difference form is

$$\frac{\Delta\rho}{\Delta t} = \left[\frac{r_{i-1}\rho_{i-1}u_{i-1}}{r_{i-1/2}\Delta r} - \frac{r_i\rho_i u_i}{r_{i-1/2}\Delta r} + \frac{\rho_{j-1}v_{j-1} - \rho_j v_j}{\Delta z}\right] \quad \text{(A8)}$$

This can be rewritten to the following form, which is used by the code

$$M_k^{(n+1)} = M_k^{(n)} + \pi\Delta t\left[\left(r_i^2 - r_{i-1}^2\right)\left(\bar{v}_B\rho_B - \bar{v}_T\rho_T\right) + 2\Delta z\left(r_{i-1}\bar{u}_L\rho_L - r_i\bar{u}_R\rho_R\right)\right] \quad \text{(A9)}$$

where

$\bar{v}_B\rho_B \equiv$ mass flux across bottom face of cell K

$\bar{v}_T\rho_T \equiv$ mass flux across top face of cell K

$\bar{u}_L\rho_L \equiv$ mass flux across left face of cell K

$\bar{u}_R\rho_R \equiv$ mass flux across right face of cell K

and where the densities are those of the donor cells and the velocities are

$$\bar{v} = \frac{1/2(\tilde{v}_M + \tilde{v}_K)}{1 + (\tilde{v}_M - \tilde{v}_K)\frac{\Delta t}{\Delta z}}$$ where M is the cell above cell K

$$\bar{u} = \frac{1/2(\tilde{u}_M + \tilde{u}_K)}{1 + (\tilde{u}_M - \tilde{u}_K)\frac{\Delta t}{\Delta r}}$$ where M is the cell to right of cell K

The velocity-weighting scheme used above is one that ensures stability in regions behind a shock front.

When mass transport is being calculated for a given cell, the transfer at the left and bottom faces is already available from the previous column sweep and cell below, respectively. Therefore, for a particular cell, the mass transport is calculated for the top and right faces only. After the masses transferred at all faces of a cell have been determined, the momenta and the total specific energy associated with these masses are computed. Then, by conserving both axial and radial momentum, the new velocities are calculated. The total energy carried by the transported masses is subtracted from the donor cell and added to the receiver cell. Then, by conserving both axial and radial momentum the new velocities are calculated. The total energy carried by the transported masses is subtracted from the donor cell and added to the receiver cell. Then, by conserving total energy the new internal energy of each cell is the difference between the new total energy and the new kinetic energy. The new specific internal energy is the new internal energy divided by the new mass.

$$M_k^{(n+1)} = M_k^{(n)} + \Delta M_B + \Delta M_I - \Delta M_T - \Delta M_R \tag{A10}$$

$$v_k^{(n+1)} = \frac{M_k^{(n)}\tilde{v}_k + \Delta M_B\tilde{v}_M + \Delta M_L\tilde{v}_M - \Delta M_T\tilde{v}_M - \Delta M_R\tilde{v}_M}{M_k^{(n+1)}} \tag{A11}$$

$$u_k^{(n+1)} = \frac{M_k^{(n)}\tilde{u}_k + \Delta M_B\tilde{u}_M + \Delta M_L\tilde{u}_M - \Delta M_T\tilde{u}_M - \Delta M_R\tilde{u}_M}{M_k^{(n+1)}} \tag{A12}$$

where M designates the donor cell.

$$I_k^{(n+1)} = \Big\{ M_k^{(n)}\left[\tilde{I}_k + \tfrac{1}{2}\left(\tilde{u}_k^2 + \tilde{v}_k^2\right)\right] + \Delta M_B\left[\tilde{I}_M + \tfrac{1}{2}\left(\tilde{u}_M^2 + \tilde{v}_M^2\right)\right]$$
$$+ \Delta M_L\left[\tilde{I}_M + \tfrac{1}{2}\left(\tilde{u}_M^2 + \tilde{v}_M^2\right)\right] - \Delta M_T\left[\tilde{I}_M + \tfrac{1}{2}\left(\tilde{u}_M^2 + \tilde{v}_M^2\right)\right]$$
$$- \Delta M_R\left[\tilde{I}_M + \tfrac{1}{2}\left(\tilde{u}_M^2 + \tilde{v}_M^2\right)\right]\Big\} \Big/ \left\{ M_K^{(n+1)} - \frac{\frac{1}{2}\left[u^{(n+1)}\right]^2 - \frac{1}{2}\left[v^{(n+1)}\right]^2}{M_k^{(n+1)}} \right\} \tag{A13}$$

where ΔM_B, ΔM_L, ΔM_T, and ΔM_R are the masses, in grams transferred across the bottom, left, top, and right faces of cell K, respectively, and where the quantities enclosed in brackets and subscripted by M are the specific total energies of the donor cells. The above are the final values of velocity, mass, and specific internal energy for the cycle. If any mass leaves the top, right, or bottom boundaries of the grid,

that mass and the associated energy are subtracted from the total mass and energy of the system respectively.

The calculation of mass transport in SHELL-OIL has another feature that removes preferential transfer caused by initial choice of indexing. If the mass out the top and right would remove more than the mass in the cell, the code recalculates new mass transfer by a weighting procedure. The ΔM_R would be its fraction of the total mass out times the mass of the cell and ΔM_T would be its fraction of the total mass out times the mass of the cell. That is,

$$\Delta M_T = \frac{\Delta M_T}{\Delta M_T + \Delta M_R}\left(M_K^{(n)} + \Delta M_L + \Delta M_B\right) \qquad \text{(A14)}$$

$$\Delta M_R = \frac{\Delta M_R}{\Delta M_T + \Delta M_R}(M_K^{(n)} + \Delta M_L + \Delta M_B) \qquad \text{(A15)}$$

In this calculation ΔM_L and ΔM_B are considered zero if they are flowing into cell K so that the maximum mass that could flow out the top and right is $M_K^{(n)}$. However, if ΔM_L and ΔM_B are flowing out of cell K, the cell is committed to deliver this mass and so the maximum mass that can flow out the top and right is $M_K^{(n)} + \Delta M_L + \Delta M_B$ (where ΔM_L and ΔM_B are negative signifying flow in negative direction).

STABILITY OF SHELL CODES

The Eulerian equations used in the first phase of the calculation are unstable, since they do not contain the dissipative mechanism necessary for a finite difference technique

to calculate shocks. Since shocks appear as mathematical surfaces on which such fluid properties as density, pressure, internal energy, and entropy have discontinuities, suitable boundary conditions (those provided by the Rankine-Hugoniot equations) are needed to connect the values of the above quantities on both sides of the shock. However, Von Neumann and Richtmeyer (6) have shown that the hydrodynamic equations can be straightforwardly solved by numerical methods if an artificial dissipative term is introduced. This term smears the shock so that the surface of discontinuity is replaced by a thin layer in which the above quantities vary rapidly but continuously. Hence, the numerical calculation proceeds as if there were no shock at all and at the same time satisfying the Rankine-Hugoniot conditions.

The instability of the calculations in the first phase introduces the main source of entropy loss in the entire calculation. However, stable calculations can be made by SHELL, since errors introduced in the second phase have the effect of smoothing out the discontinuities.

Harlow (15) has shown that the treatment of mass movement in the second phase of the calculation produces dissipative effects that give stability to the calculation. He demonstrated this by expanding the difference equations in a Taylor series about some central space and time. The result was the original differential equations plus some additional terms that, in lowest order, have the appearance of true

viscous and heat conduction terms. Hence they are called the effective viscosity and effective heat conduction.

Conservation of Mass

$$\frac{\partial \rho}{\partial t} + \frac{\partial (r\rho u)}{r\partial r} + \frac{\partial (\rho v)}{\partial z} = 0 \qquad \text{(A16)}$$

Conservation of Momentum--r direction

$$\rho\left(\frac{\partial u}{\partial t} + u\,\frac{\partial u}{\partial r} + v\,\frac{\partial u}{\partial z}\right) + \frac{\partial p}{\partial r} = \frac{\partial}{r\partial r}\left(r\lambda_u\,\frac{\partial u}{\partial r}\right) + \frac{\partial}{\partial z}\left(\lambda_v\,\frac{\partial u}{\partial z}\right) \qquad \text{(A17)}$$

Conservation of Momentum--z direction

$$\rho\left(\frac{\partial v}{\partial t} + u\,\frac{\partial v}{\partial r} + v\,\frac{\partial v}{\partial z}\right) + \frac{\partial p}{\partial z} = \frac{\partial}{r\partial r}\left(r\lambda_u\,\frac{\partial v}{\partial r}\right) + \frac{\partial}{\partial z}\left(\lambda_v\,\frac{\partial v}{\partial z}\right) \qquad \text{(A18)}$$

Conservation of Energy

$$\rho\left(\frac{\partial I}{\partial t} + u\,\frac{\partial I}{\partial r} + v\,\frac{\partial I}{\partial z}\right) + p\left[\frac{\partial (ru)}{r\partial r} + \frac{\partial v}{\partial z}\right] = \frac{\partial}{r\partial r}\left(r\lambda_u\,\frac{\partial I}{\partial r}\right)$$

$$+ \frac{\partial}{\partial z}\left(\lambda_v\,\frac{\partial I}{\partial z}\right) + \lambda_u\left[\left(\frac{\partial u}{\partial r}\right)^2 + \left(\frac{\partial v}{\partial r}\right)^2\right]$$

$$+ \lambda_v\left[\left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2\right] \qquad \text{(A19)}$$

where

$$\lambda_u = 1/2\rho|u|\Delta r$$

$$\lambda_v = 1/2\rho|v|\Delta z$$

However, Bjork (40) has shown that any attempt to relate the terms appearing on the right-hand side of the equations above to distinct physical effects leads to many contradictions--the "effective viscosity" terms contain symmetric and

antisymmetric elements; the origin of the symmetric elements is not viscous; the presence of the antisymmetric elements simply adds confusion to any attempted representation; the terms do not leave the flow equations in an invariant form; a heat conduction law is implied where heat flow is proportional to the gradient of specific internal energy rather than of temperature. Moreover, the values of these terms far exceed the magnitude of the analogous physical terms. Therefore, these terms cannot be thought of an anything except errors, but they do make SHELL a practical code.

The form of the effective viscosity suggests another limitation of the SHELL code. Since the viscous effects are proportional to the local mean fluid speed, in regions of low fluid speed the viscosity will become ineffective and the instabilities of the difference equation will cause an exponential growth of any perturbations to the solution. But as the velocities increase and become comparable to the local sound speed, the viscosity again takes effect to limit further growth of instability. Hence, the instability is bounded and calculations can be made. This effective viscosity is the main source of entropy production in the calculation.

A later modification to the SHELL codes, accomplished by Nawrocki, included an artificial viscosity addition scheme. By selecting appropriate values for three parameters, S1, S2 and S3, an energy dissipating effect is obtained associated with the sound speed, the velocity gradient and the magnitude of the velocity, respectively. By proper selection of the

parameters, the effective viscosity inherent in SHELL can be enhanced or diminished.

DEVELOPMENT OF SHELLTC

In developing a version of SHELL for use in cartesian coordinates we return to the basic equations for 2-D inviscid compressible flow in that coordinate system.

Mass

$$\left(\frac{\partial}{\partial t} + \bar{U}\cdot\nabla\right)\rho + \rho\nabla\cdot\bar{U} = 0 \qquad \text{(A20)}$$

Momentum

$$\rho\left(\frac{\partial}{\partial t} + \bar{U}\cdot\nabla\right)\bar{U} + \nabla p + \rho\nabla\phi = 0 \qquad \text{(A21)}$$

Energy

$$\rho\left(\frac{\partial}{\partial t} + U\cdot\nabla\right)E + \nabla\cdot p\bar{U} + \rho\bar{U}\cdot\nabla\phi = 0 \qquad \text{(A22)}$$

State

$$p = p(P,I) \qquad \text{(A23)}$$

Neglecting $\nabla\phi$ and holding all items constant with respect to Z, we have, in cartesian coordinates,

$$\bar{U}\cdot\nabla = U_x \frac{\partial}{\partial x} + U_y \frac{\partial}{\partial y} \qquad \text{(A24)}$$

$$\nabla\cdot\bar{U} = \frac{\partial U_x}{\partial x} + \frac{\partial U_y}{\partial y} \qquad \text{(A25)}$$

Using equations A24 and A25; equations A20, A21 and A22 become (letting $U_x = u$, $U_y = v$)

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) = 0 \tag{A26}$$

$$\rho \frac{\partial u}{\partial t} + \rho u \frac{\partial u}{\partial x} + \rho v \frac{\partial u}{\partial y} + \frac{\partial P}{\partial x} = 0 \tag{A27}$$

$$\rho \frac{\partial v}{\partial t} + \rho u \frac{\partial v}{\partial x} + \rho v \frac{\partial v}{\partial y} + \frac{\partial P}{\partial y} = 0 \tag{A28}$$

$$\rho \frac{\partial E}{\partial t} + \rho u \frac{\partial E}{\partial x} + \rho v \frac{\partial E}{\partial y} + \frac{\partial}{\partial x}(Pu) + \frac{\partial}{\partial y}(Pv) = 0 \tag{A29}$$

Dropping the convective terms as was done previously with axisymmetric coordinates we get from equations A27 through A29

$$\frac{\partial u}{\partial t} + \frac{1}{\rho} \frac{\partial P}{\partial x} = 0 \tag{A30}$$

$$\frac{\partial v}{\partial t} + \frac{1}{\rho} \frac{\partial P}{\partial y} = 0 \tag{A31}$$

$$\frac{\partial I}{\partial t} + \frac{P}{\rho}\left[\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right] = 0 \tag{A32}$$

(where E has been replaced by I). In finite difference form equations A30 and A31 are the same as was obtained for the axisymmetric case earlier. They are

$$\tilde{u}_k = u_k^{(n)} + \frac{1}{\rho_k^{(n)}}\left[\frac{PL^{(n)} - PR^{(n)}}{\Delta x}\right]\Delta t \tag{A33}$$

$$\tilde{v}_k = v_k^{(n)} + \frac{1}{\rho_k^{(n)}}\left[\frac{PB^{(n)} - PA^{(n)}}{\Delta y}\right]\Delta t \tag{A34}$$

where

$$PL = \frac{P_k + P_{kL}}{2}, \qquad PR = \frac{P_k + P_{kR}}{2}$$

The energy equation A32 takes a somewhat different form in cartesian coordinates than in axisymmetric. In finite difference form it becomes

$$\frac{\tilde{I}_k - I_k^{(n)}}{\Delta} = \frac{-P_k^{(n)}}{\rho_k^{(n)}}\left[\frac{\frac{\left(\tilde{u}_{kR} + \tilde{u}_{kR}^{(n)}\right)}{2} - \frac{\left(\tilde{u}_{kL} + \tilde{u}_{kL}^{(n)}\right)}{2}}{2\Delta x}\right.$$

$$\left. + \frac{\frac{\left(\tilde{v}_{kA} + \tilde{v}_{kA}^{(n)}\right)}{2} - \frac{\left(\tilde{v}_{kB} + \tilde{v}_{kB}^{(n)}\right)}{2}}{2\Delta y}\right]$$

Here again the velocity at each point is the average of that at the beginning of the time step and the interim (~) value calculated above in equations A31 and A34. Now let

$$VB = \frac{v_k + v_{kB}}{2}, \qquad VA = \frac{v_k + v_{kA}}{2}$$

$$UL = \frac{u_k + u_{kL}}{2}, \qquad UR = \frac{u_k + u_{kR}}{2} \tag{A35}$$

Then

$$VB^{(n)} - VA^{(n)} + \tilde{V}B - \tilde{V}A$$

$$= \frac{v_k^{(n)} + v_{kB}^{(n)} - v_k^{(n)} - v_{kA}^{(n)} + \tilde{v}_k + \tilde{v}_{kB} - \tilde{v}_k - \tilde{v}_{kA}}{2}$$

$$= \frac{\left(v_{kB}^{(n)} + \tilde{v}_{kB}\right) - \left(v_{kA}^{(n)} + \tilde{v}_{kA}\right)}{2} \qquad \text{(A36)}$$

and

$$UL^{(n)} - UR^{(n)} + \tilde{U}L - \tilde{U}R = \left(u_{kL}^{(n)} + \tilde{u}_{kL}\right) - \left(u_{kR}^{(n)} + \tilde{u}_{kR}\right) \qquad \text{(A37)}$$

With equations A36 and A37, equation A33 can be rewritten

$$\tilde{I}_k = I_k^{(n)} + \frac{P_k^{(n)}}{\rho_k^{(n)}}\left[\frac{VB^{(n)} - VA^{(n)} + \tilde{V}B - \tilde{V}A}{\Delta y}\right.$$

$$\left. + \frac{UL^{(n)} - UR^{(n)} + UL - UR}{\Delta x}\right]\frac{\Delta t}{2} \qquad \text{(A38)}$$

This equation is now accomplished by two passes through the code--the first computes the change in energy in time $\Delta t/2$ using the old (n) values of velocities and the second computes the changes in time $\Delta t/2$ using the new velocities (~) computed above by equations A33 and A34.

In the phase II calculation mass is permitted to flow governed by the continuity equation. In finite difference form, equation A26 becomes

$$\frac{\Delta\rho}{\Delta t} = \left[\frac{\rho_{i-1}u_{i-1} - \rho_i u_i}{\Delta x} + \frac{\rho_{j-1}v_{j-1} - \rho_j v_j}{\Delta y}\right] \qquad \text{(A39)}$$

$$\rho_k^{(n+1)} = \rho_k^{(n)} + \left[\frac{\rho_{i-1}u_{i-1} - \rho_i u_i}{\Delta x} + \frac{\rho_{j-1}v_{j-1} - \rho_j v_j}{\Delta y}\right]\Delta t \qquad \text{(A40)}$$

We again use the same velocity weighting scheme as with axi-symmetric coordinates

$$\text{for } v_{j-1} \text{ we use } \bar{V}_B = \frac{1/2\ (\tilde{v}_{kB} + \tilde{v}_k)}{1 + (\tilde{v}_{kB} + \tilde{v}_k)\frac{\Delta t}{\Delta y}}$$

$$\text{for } v_j \text{ we use } \bar{V}_T = \frac{1/2\ (\tilde{v}_{kT} + \tilde{v}_k)}{1 + (\tilde{v}_{kT} + \tilde{v}_k)\frac{\Delta t}{\Delta y}} \qquad \text{(A41)}$$

$$\text{for } u_{i-1} \text{ we use } \bar{U}_L = \frac{1/2\ (\tilde{u}_{kL} + \tilde{u}_k)}{1 + (\tilde{u}_{kL} + \tilde{u}_k)\frac{\Delta t}{\Delta x}}$$

$$\text{for } u_i \text{ we use } \bar{U}_R = \frac{1/2(\tilde{u}_{kR} + \tilde{u}_k)}{1 + (\tilde{u}_{kR} + \tilde{u}_k)\frac{\Delta t}{\Delta x}}$$

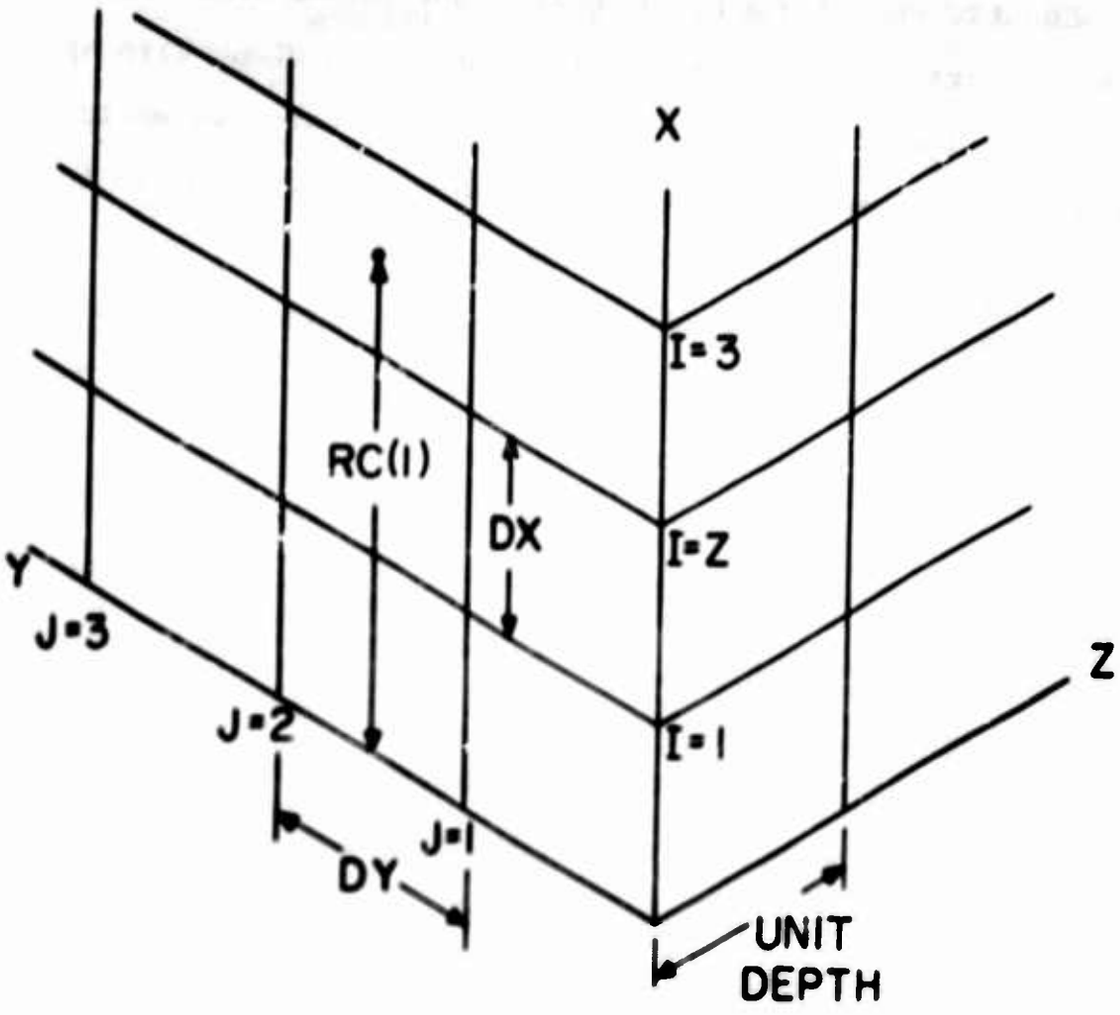
Using these and multiplying by the volume of a cell (where volume = $\Delta x \cdot \Delta y$) equation A40 becomes

$$\rho_k^{(n+1)}\Delta x\Delta y = \rho_k^{(n)}\Delta x\Delta y + [\Delta x(\rho_B\bar{V}_B - \rho_T\bar{V}_T) + \Delta y(\rho_L\bar{U}_L - \rho_R\bar{U}_R)]\Delta t \qquad \text{(A42)}$$

Finally

$$M_k^{(n+1)} = M_k^{(n)} + \Delta t[\Delta x(\rho_B\bar{V}_B - \rho_T\bar{V}_T) + \Delta y(\rho_L\bar{U}_L - \rho_R\bar{U}_R)] \qquad \text{(A43)}$$

The computational grid for SHELLTC is constructed in the XY plane and is symmetric about the y axis as depicted in Figure A3. It takes its 3-D form by including a unit depth in the Z direction.



RC(I) is the X distance to the center of a cell in the Ith column.

Y dimension of each cell is DY.

X dimension of each cell is DX

Cross-sectional area of each cell is TAU(I) where

TAU(I) = DX·Unit Depth = DX

Volume of each cell is

VOL(I) = DX·DY

Figure A1. SHELL Zoning--Two-Dimensional Planar Case

SHELLTC was formed by including the hydrodynamic equation for cartesian coordinates into SHELL OIL along with appropriate flags for determining which coordinate system is being used. Some modification of CLAM was also required to obtain CLAMTC, the generator code for SHELLTC.

The effective viscosity, inherent in SHELL, is continued in SHELLTC as is the Nawrocki artificial viscosity scheme described above. Consideration was given to reducing the effective viscosity inherent in SHELLTC by a proper selection of the parameters S1, S2 and S3 in hopes of sharpening the shock front produced by a SHELLTC calculation. For this reason a parametric study was performed using a 1-D planar blast wave as a trial problem (similar to Reference 32) to determine the effect of variations in those S parameters. The shape of the shock front obtained for selected values of S1, S2 and S3 are compared with a 1-D similarity solution for the planar blast wave in Figures A4 through A7. The results of this study indicated that the closest reproduction of the analytical solution was obtained with the following values:

$S_1 = -0.1$

$S_2 = 0.1$

$S_3 = -0.4$

Although these values worked satisfactorily for problems where the deposited energy was small, difficulty was

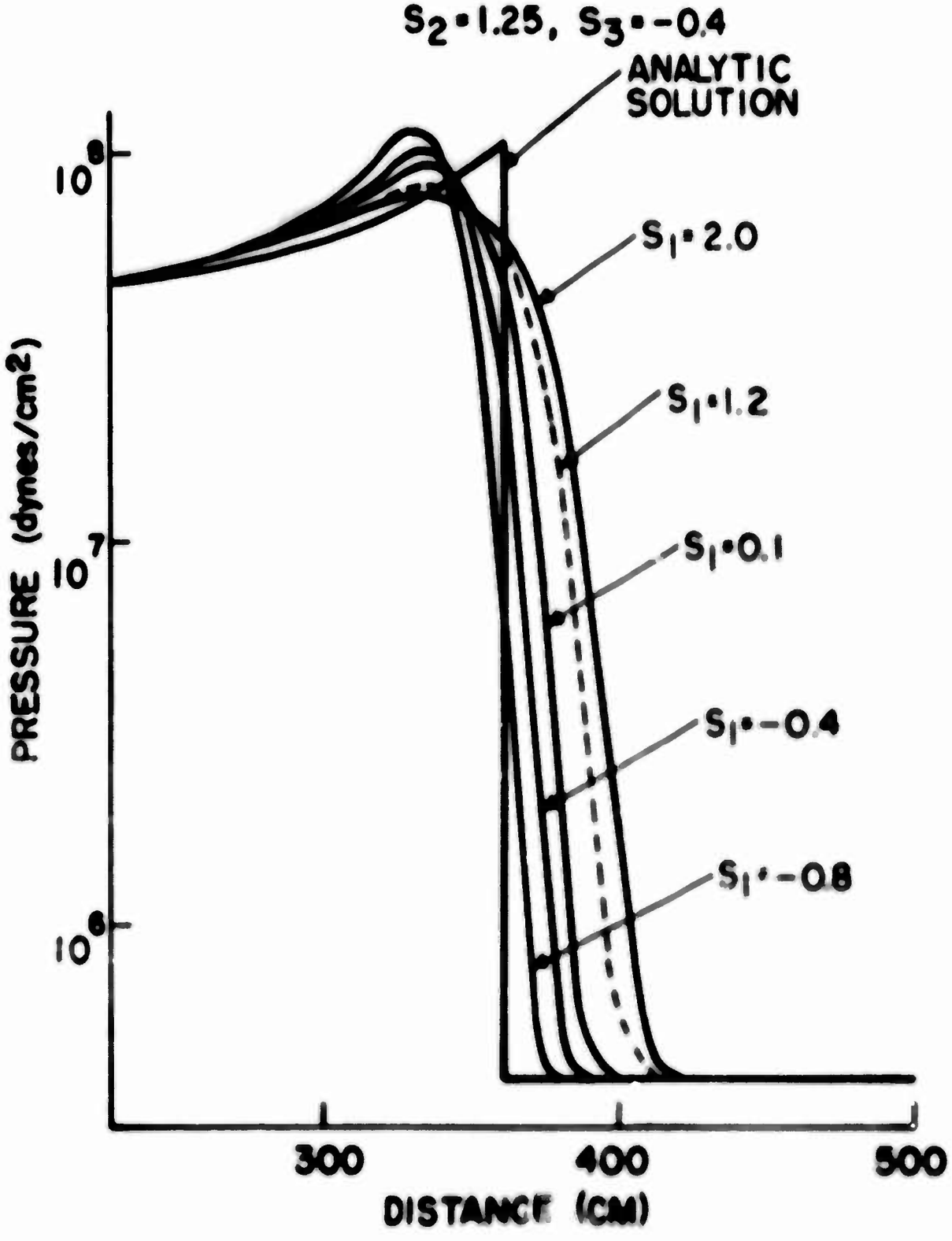


Figure A4. Variation [illegible] parameter S_1

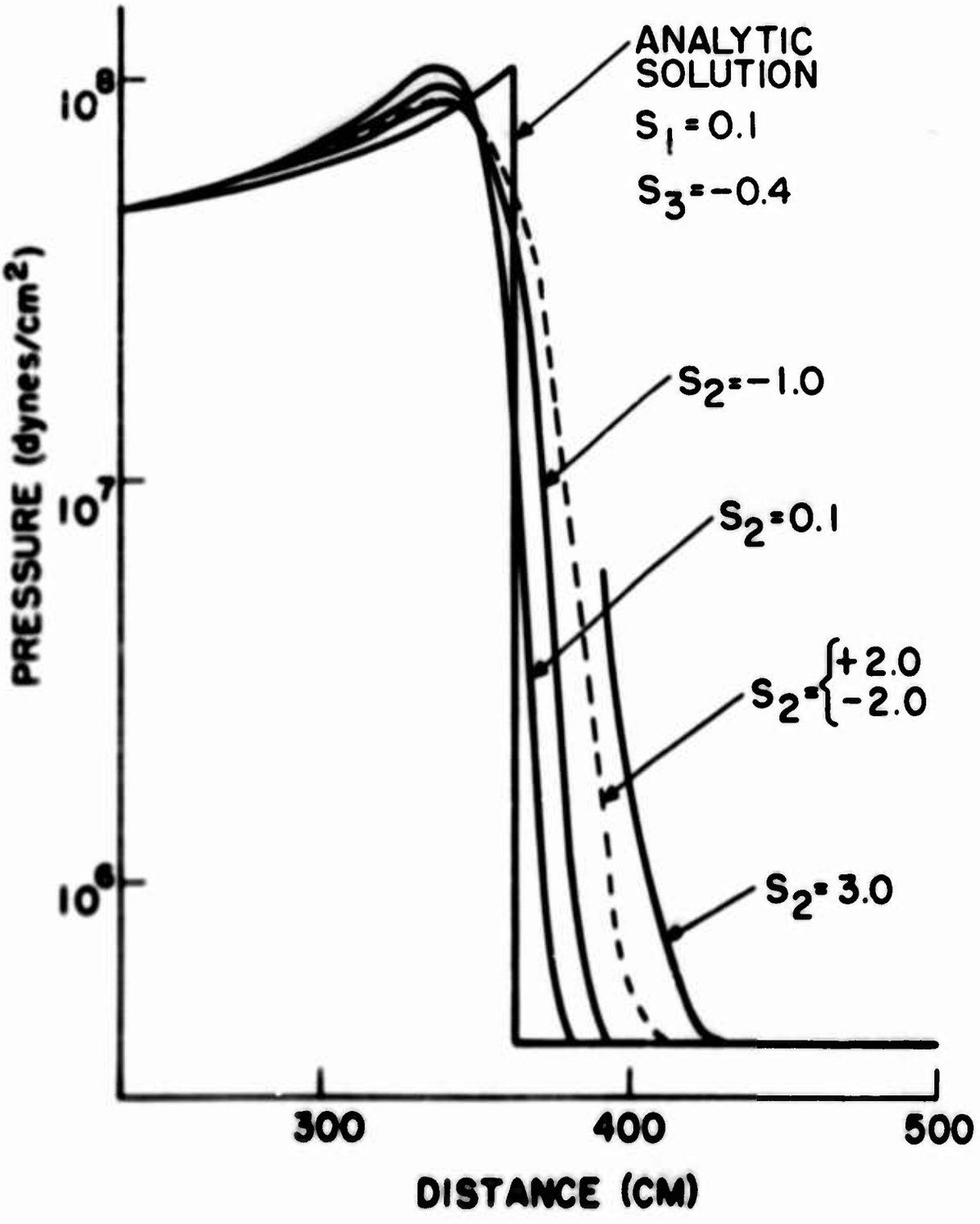


Figure A5. Variation of Parameter S_2

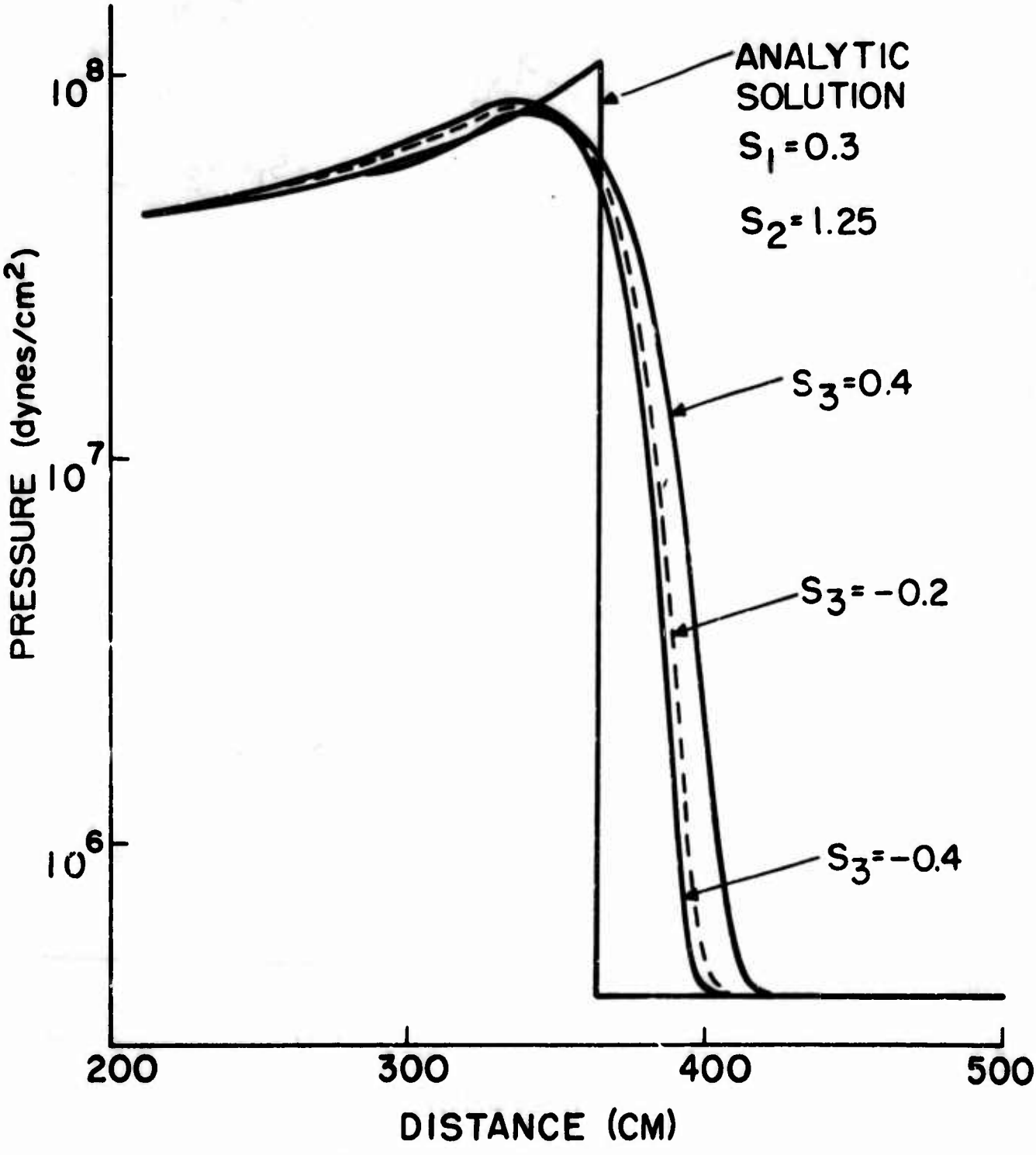


Figure A6. Variation of Parameter S_3

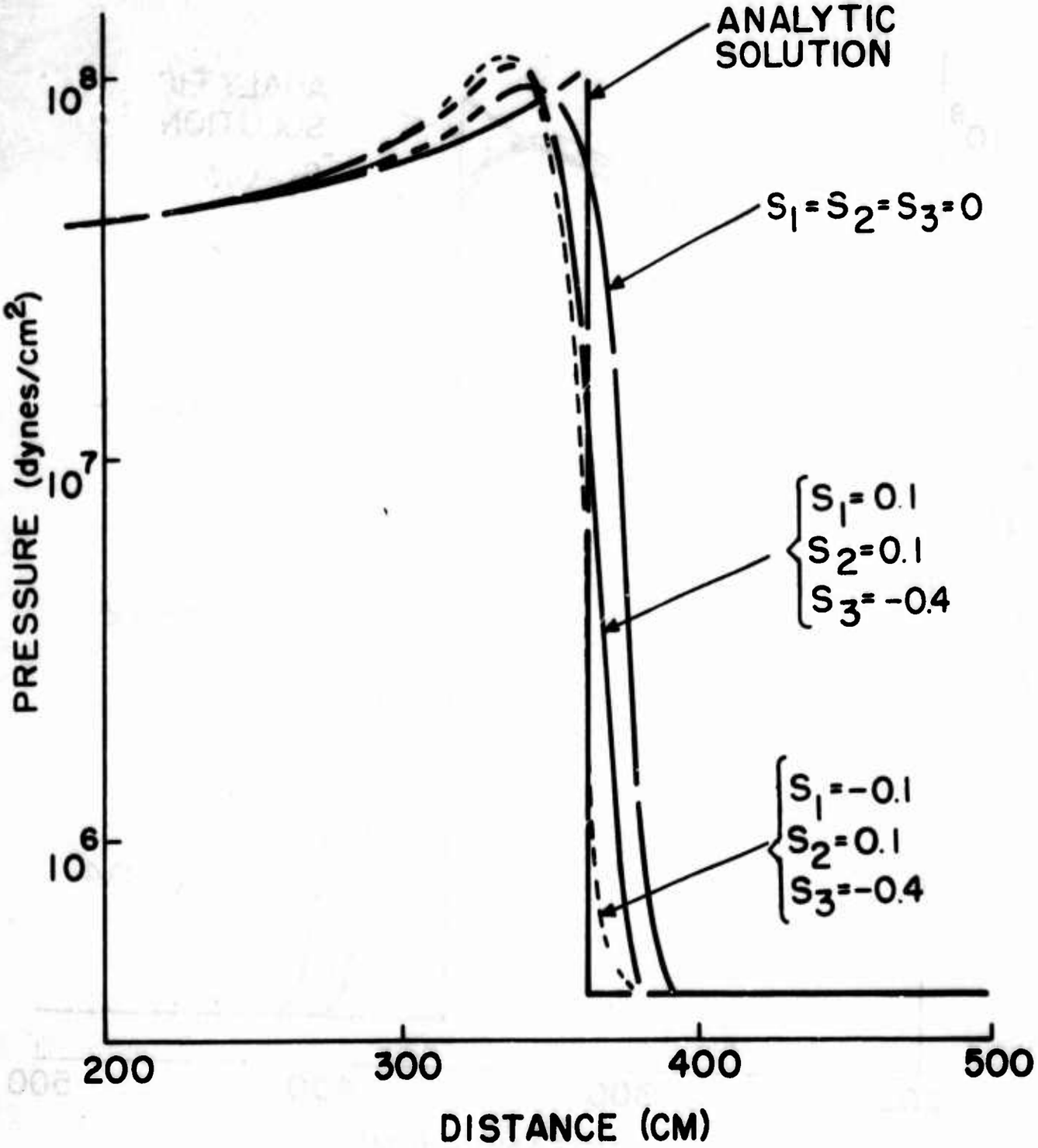


Figure A7. Results of S Parameter Study

encountered during large energy deposition. Excessive computational time was required and in some instance the computation became unstable. For this reason, the three S parameters were set to zero (eliminating, in effect, the Nawrocki artificial viscosity scheme) during all calculations performed for this dissertation. Hence, the shock structure obtained throughout the study is that provided by the effective viscosity inherent in SHELLTC.

An energy deposition scheme, described in the main text, was incorporated into SHELLTC in order to solve the 2-D blast wave problem of interest. The code in its present configuration is suitable for use with the CDC-3400 or CDC-6600 computer. A complete listing of SHELLTC, as used in this work, follows.

LISTING OF SHELLTC CODE

```
      PROGRAM SHELLTC(INPUT,OUTPUT,TAPE4,TAPE7,TAPE9,TAPE5=INPUT,
     1  TAPE6=OUTPUT)
C ***** REDUCED FOR CDC-3400 OPERATION
CMAIN
C
C     ****** NOTE 1 MATERIAL ONLY (X)) ************
      COMMON Z(19000)
      CALL INPUT
   10 CALL CDT
      CALL EDIT
      IF(SENSE LIGHT 1)30,20
   20 CALL PH1
      CALL PH2
      GO TO 10
   30 CALL EXIT
      END
```

```
      SUBROUTINE CARDS
      DIMENSION TABLE(1),CARD(7),LABLE(1)
      COMMON           TABLE
      EQUIVALENCE(TABLE(1),LABLE(1))
      WRITE (6,10)
    1 READ  (5,11)IEND,LOC,NUMWPC,(CARD(I),I=1,NUMWPC)
      WRITE (6,12)IEND,LOC,NUMWPC,(CARD(I),I=1,NUMWPC)
      DO 4 I=1,NUMWPC
      J=LOC+I-1
      IF(IEND-2)2,5,2
    5 LABLE(J)=XFIXF(CARD(I))
      GO TO 4
    2 TABLE(J)=CARD(I)
    4 CONTINUE
      IF(IEND-1)1,3,1
    3 RETURN
C               FORMATS
   10 FORMAT(20H1SHELL    INPUT CARDS///)
   11 FORMAT(I1,I5,I1,0P7E9.4)
   12 FORMAT(1H I4,I7,I3,1P7E14.6)
      END
```

```
      SUBROUTINE INPUT
C                  D   I   M   E   N   S   I   O   N
      DIMENSION AIX(3500),AMX(3500),P(3500),
     1 U(3500),V(3500),X(60),Y(100),Z(150),
     2 IZ(150),TAU(60),PL(100),UL(100),PR(100),
     3 UR(100),IW1(15),FLEFT(100),YAMC(100),
     4 SIGC(100),GAMC(100)
C
C                  C       O       M       M       O       N
C
      COMMON        Z       ,XO     ,X      ,Y
      COMMON        AIX     ,AM     ,AMX    ,AREA   ,E
      COMMON        FS      ,I      ,II     ,IR     ,IWS
      COMMON        J       ,K      ,KP     ,L      ,M
      COMMON        MZ      ,N      ,NK     ,NO     ,NOP
      COMMON        NR      ,P      ,PABOVE ,PBLO   ,PIDTS
      COMMON        PRR     ,PC     ,REZ    ,RHO    ,RR
      COMMON        SIG     ,TAU    ,TAUDTS ,TTX    ,TTY
      COMMON        U       ,URR    ,UT     ,UTEF   ,UU
      COMMON        UVMAX   ,V      ,VABOVE ,VBLO   ,VEL
      COMMON        VTEF    ,WS     ,WSA    ,WSB    ,WSC
      COMMON        XLF     ,XR     ,YLW
      COMMON        YU      ,UR     ,PR
C
C                  E   Q   U   I   V   A   L   E   N   C   E
C
     0EQUIVALENCE       (Z,IZ,PPOS),    (Z(2),CYCLE),   (Z(3),DT),
     1(Z(4),PRINTS),    (Z(5),PRINTL),  (Z(6),DUMPT),   (Z(7),CSTOP),
     2(Z(8),PIDY),      (Z(9),TMZ),     (Z(10),GAM),    (Z(11),GAMD),
     3(Z(12),GAMX),     (Z(13),ETH),    (Z(14),FFA),    (Z(15),FFB),
     4(Z(16),TMDZ),     (Z(17),TMXZ),   (Z(18),XMAX),   (Z(19),TXMAX),
     5(Z(20),TYMAX),    (Z(21),AMDM),   (Z(22),AMXM),   (Z(23),DNN),
     6(Z(24),DMIN),     (Z(25),FEF),    (Z(26),DTNA),   (Z(27),CVIS),
     7(Z(28),NPR),      (Z(29),NPRI),   (Z(30),NC),     (Z(31),NPC),
     8(Z(32),NRC),      (Z(33),IMAX),   (Z(34),IMAXA),  (Z(35),JMAX),
     9(Z(36),JMAXA),    (Z(37),KMAX),   (Z(38),KMAXA),  (Z(39),NMAX)
     0EQUIVALENCE       (Z(40),ND),     (Z(41),KDT),    (Z(42),IXMAX),
     1(Z(43),NOD),      (Z(44),NOPP),   (Z(45),NIMAX),  (Z(46),NJMAX),
     2(Z(47),I1),       (Z(48),I2),     (Z(49),I3),     (Z(50),I4),
     3(Z(51),N1),       (Z(52),N2),     (Z(53),N3),     (Z(54),N4),
     4(Z(55),N5),       (Z(56),N6),     (Z(57),N7),     (Z(58),N8),
     5(Z(59),N9),       (Z(60),N10),    (Z(61),N11),    (Z(62),NRM),
     6(Z(63),TRAD),     (Z(64),XNRC),   (Z(65),SN),     (Z(66),DXN),
     7(Z(67),RADER),    (Z(68),RADET),  (Z(69),RADER),  (Z(70),DTRAD),
     8(Z(71),REZECT),   (Z(72),PSTOP),  (Z(73),SHELL),  (Z(74),PBOUND)
     9(Z(75),TOZONE),   (Z(76),ECK),    (Z(77),SBOUND), (Z(78),X1)
     0EQUIVALENCE       (Z(79),X2),     (Z(80),Y1),     (Z(81),Y2),
     1(Z(82),CABLN),    (Z(83),VISC),   (Z(84),T),      (Z(85),GMAX),
     2(Z(86),WSGD),     (Z(87),WSGX),   (Z(88),GMADR),  (Z(89),GMAXR),
     3(Z(90),S1),       (Z(91),S2),     (Z(92),S3),     (Z(93),S4),
     4(Z(94),S5),       (Z(95),S6),     (Z(96),S7),     (Z(97),S8),
     5(Z(98),S9),       (Z(99),S10),
     6(Z(127),DX),      (Z(128),DY1),   (Z(129),DY),
     7(Z(130),NFITB),   (Z(131),NFITT), (Z(132),NFITR),
     8(Z(133),NPARB),   (Z(134),NPART), (Z(135),NPARR)
C
      EQUIVALENCE       (P,IW1)
      EQUIVALENCE(P(200),GAMC),(P(400),YAMC)
      EQUIVALENCE       (UP,UL,FLEFT),  (PR,PL,SIGC)
```

```
C
      EQUIVALENCE  (Z(120),DETVEL),  (Z(122),EXEN)
      EQUIVALENCE  (Z(123),EXRHO),  (Z(124),THICK)
      SENSE LIGHT 3
      DO 300 I=1,14999
  300 Z(I)=0.
C
      READ  (5,8004)IWS
      WRITE (6,8004)IWS
    6 CALL CARDS
C
C              READ TAPE
C
      STOR1 = Z(235)
      STOR2 = Z(236)
 1000 MZ=150
      IWS=0
      REWIND 7
 1003 REWIND 4
 1004 READ(4)PR(1),PR(2),N3
      WRITE (6,8800) PR(1),PR(2),N3
 8800 FORMAT (10X,15HFIRST RECORD IS ,2E12.4,I6,/)
      NR=N3+5
 1006 IF(PR(1)-555.0)1010,1016,1010
 1010 IF(PR(1).EQ.666.)1012,1013
 1012 DO 1014 I=1,5
 1014 BACKSPACE 4
      IF (GAMD) 1023,1023,1015
 1015 DO 1017 I = 1,4
 1017 BACKSPACE 7
      GO TO 1023
 1013 IWS=IWS+1
      WRITE (6,8801) IWS
 8801 FORMAT (10X,6HIWS = ,I2,/)
 1011 IF(XMODF(IWS,3))9902,9902,1003
 1016 WRITE (6,8881) Z(235), Z(236), PR(2), PROB
 8881 FORMAT (10X,9HZ(235) = ,E14.6,12H   Z(236) = ,E14.6,
     1  11H   PR(2) = ,E14.6,10H   PROB = ,E14.6,/)
      IF (Z(236)-PR(2)) 1025,1025,1019
 1019 IF (GAMD) 1020,1020,1021
 1020 READ(4) DUM
      READ(4) DUM
      READ(4) DUM
      READ(4) DUM
      GO TO 1004
 1021 WRITE (7) PR(1),PR(2),N3
      READ(4)(Z(I),I=1,MZ)
      READ(4)(U(I),V(I),AMX(I),AIX(I),P(I),I=1,KMAXA)
      READ(4)X(0),(X(I),TAU(I),I=1,IMAX)
      READ(4)(Y(I-1),I=1,JMAXA)
      WRITE(7)(Z(I),I=1,MZ)
      WRITE(7)(U(I),V(I),AMX(I),AIX(I),P(I),I=1,KMAXA)
      WRITE(7)X(0),(X(I),TAU(I),I=1,IMAX)
      WRITE(7)(Y(I-1),I=1,JMAXA)
      WRITE (6,8805) PR(2)
      Z(235) = STOR1
      Z(236) = STOR2
      GO TO 1004
```

```
 1025 IF (GAMD) 1023,1023,1022
 1022 WRITE (7) PR(1),PR(2),N3
 1023 READ(4)(Z(I),I=1,MZ)
      WRITE (6,8802) (Z(I),I = 1,3)
 8802 FORMAT (1HX,25HTHE FIRST 3 Z VALUES ARE ,3E12.6,/)
      IF(ABSF(PROB-Z(235))-.01)1024,1024,9901
 1024 READ(4)(U(I),V(I),AMX(I),AIX(I),P(I),I=1,KMAXA)
      READ(4)X(0),(X(I),TAU(I),I=1,IMAX)
      READ(4)(Y(I-1),I=1,JMAXA)
 1034 READ(4)FS,RC,RR
      IF (GAMD) 1030,1030,1031
 1031 WRITE(7)(Z(I),I=1,MZ)
      WRITE(7)(U(I),V(I),AMX(I),AIX(I),P(I),I=1,KMAXA)
      WRITE(7)X(0),(X(I),TAU(I),I=1,IMAX)
      WRITE(7)(Y(I-1),I=1,JMAXA)
      WRITE(7)FS,RC,RR
      WRITE (6,8805) PR(2)
 1030 WRITE (6,8803)FS,RC,RR
 8803 FORMAT (10X,18HFS, RC AND RR ARE ,3E12.6,/)
  501 CONTINUE
 1036 IF(FS-555.0)9904,1040,1038
 1038 IF(RC-666.0)9905,1040,9905
 1040 GO TO 10
C
C**** END OF READ TAPE ***********************************************
C
C                READ IN REMAINING INPUT CARDS
C
   10 CONTINUE
      CALL CARDS
   40 DO 45 K=1,KMAXA
   45 P(K)=0.0
      T=T-DTNA
      NC=NC-1
      CYCLE=NC
      NPC=NPC-1
      UVMAX=0.0
      JX=MZ-4
   62 DO 80 I=1,JX,4
      K=I+3
      DO 65 J=I,K
      IF(Z(J))70,65,70
   65 CONTINUE
      GO TO 80
   70 CONTINUE
      WRITE (6,8111)I,(Z(L),L=I,K)
      WRITE (6,8112) I,(Z(L),L=I,K)
   80 CONTINUE
      GO TO 10000
C               ERROR
 9901 NK=1023
      GO TO 9999
 9902 NK=1011
      GO TO 9999
C
 9904 NK=1036
      GO TO 9999
 9905 NK=1038
```

```
      GO TO 9999
 9906 NK=2000
 9999 NR=1
      CALL ERRER(NR,NK)
C
10000 RETURN
C
C                FORMATS
 8000 FORMAT(7E10.3,I2)
 8805 FORMAT (7H0CYCLE ,F5.0,19H COPIED ONTO TAPE 7 //)
 80040FORMAT(I1,71H
     1                    )
 8111 FORMAT (I4,4(4X,1PE16.8))
 8112 FORMAT(I4,4025)
      END


      SUBROUTINE COT
C
C                D   I   M   E   N   S   I   O   N
C
      DIMENSION AIX(3500),AMX(3500),P(3500),
     1 U(3500),V(3500),X(60),Y(100),Z(150),
     2 IZ(150),TAU(60),PL(100),UL(100),PR(100),
     3 UR(100),IW1(15),FLEFT(100),YAMC(100),
     4 SIGC(100),GAMC(100)
C
C                C       O       M       M       O       N
C
      COMMON          Z       ,XO       ,X       ,Y
      COMMON          AIX     ,AM       ,AMX     ,AREA     ,E
      COMMON          FS      ,I        ,II      ,IR       ,IWS
      COMMON          J       ,K        ,KP      ,L        ,M
      COMMON          MZ      ,N        ,NK      ,NO       ,NOP
      COMMON          NR      ,P        ,PABOVE  ,PBLO     ,PIOTS
      COMMON          PRR     ,RC       ,REZ     ,RHO      ,RR
      COMMON          SIG     ,TAU      ,TAUDTS  ,TTX      ,TTY
      COMMON          U       ,URR      ,UT      ,UTEF     ,UU
      COMMON          UVMAX   ,V        ,VABOVE  ,VBLO     ,VEL
      COMMON          VTEF    ,WS       ,WSA     ,WSB      ,WSC
      COMMON          XLF     ,XR       ,YLW
      COMMON          YJ      ,UR       ,PR
C
C                E   Q   U   I   V   A   L   E   N   C   E
C
     0EQUIVALENCE       (Z,IZ,PROB),      (Z(2),CYCLE),     (Z(3),JT),
     1(Z(4),PRINTS),    (Z(5),PRINTL),    (Z(6),DUMPT7),    (Z(7),CSTOP),
     2(Z(8),PIDY),      (Z(9),TMZ),       (Z(10),GAM),      (Z(11),GAMO),
     3(Z(12),GAMX),     (Z(13),ETH),      (Z(14),FFA),      (Z(15),FFB),
     4(Z(16),TMDZ),     (Z(17),TMXZ),     (Z(18),XMAX),     (Z(19),TXMAX),
     5(Z(20),TYMAX),    (Z(21),AMDM),     (Z(22),AMXM),     (Z(23),DNN),
     6(Z(24),DMIN),     (Z(25),FEF),      (Z(26),DTNA),     (Z(27),CVIS),
     7(Z(28),NPR),      (Z(29),NPRI),     (Z(30),NC),       (Z(31),NPC),
     8(Z(32),NRC),      (Z(33),IMAX),     (Z(34),IMAXA),    (Z(35),JMAX),
     9(Z(36),JMAXA),    (Z(37),KMAX),     (Z(38),KMAXA),    (Z(39),NMAX)
     0EQUIVALENCE       (Z(40),NO),       (Z(41),KDT),      (Z(42),IXMAX),
     1(Z(43),NOD),      (Z(44),NOPR),     (Z(45),NIMAX),    (Z(46),NJMAX),
     2(Z(47),I1),       (Z(48),I2),       (Z(49),I3),       (Z(50),I4),
```

```
     3(Z(51),N1),      (Z(52),N2),      (Z(53),N3),      (Z(54),N4),
     4(Z(55),N5),      (Z(56),N6),      (Z(57),N7),      (Z(58),N8),
     5(Z(59),N9),      (Z(60),N10),     (Z(61),N11),     (Z(62),NMM),
     6(Z(63),TPAOI,    (Z(64),XM(6),    (Z(65),SMI,      (Z(66),DMM),
     7(Z(67),RADEG),   (Z(68),PAINT),   (Z(69),RADIUS),  (Z(70),DTRAD),
     8(Z(71),REZFCT),  (Z(72),PSTOP),   (Z(73),SMELL),   (Z(74),DDOUM),
     9(Z(75),TOZONE),  (Z(76),IC(),     (Z(77),SDOUMD),  (Z(78),Z1)
     0EQUIVALENCE      (Z(79),Z2),      (Z(80),V1),      (Z(81),V2),
     1(Z(82),CARLN),   (Z(83),VISC),    (Z(84),T),       (Z(85),GMAX),
     2(Z(86),WSGD),    (Z(87),WSGX),    (Z(88),GMAOD),   (Z(89),GMAXD),
     3(Z(90),S1),      (Z(91),S2),      (Z(92),S3),      (Z(93),S4),
     4(Z(94),S5),      (Z(95),S6),      (Z(96),S7),      (Z(97),S8),
     5(Z(98),S9),      (Z(99),S10),
     6(Z(127),DX),     (Z(128),DY),     (Z(129),DZ),
     7(Z(130),NFITR),  (Z(131),NFITT),  (Z(132),NFITR),
     8(Z(133),NPARR),  (Z(134),NPART),  (Z(135),NPARR)
C
      EQUIVALENCE       (P,IM1)
      EQUIVALENCE(P(200),GAMC),(P(400),VAMC)
      EQUIVALENCE       (UR,UL,FLEFT),   (PR,PL,SIGC)
      EQUIVALENCE (Z(131),ENSUM)
      EQUIVALENCE (Z(121),JCELL)
      EQUIVALENCE  (Z(128),DETVEL),  (Z(122),FXEN)
      EQUIVALENCE  (Z(123),FXMMO),  (Z(124),TMICC)
      EQUIVALENCE (Z(125),AJCELL)
C
C     ***************************************************
C
      DO 300 I = 1,IMAX
      K2 = I + IMAXA
      K3 = K2 + IMAX
      EO2 = (AIX(K2) + 0.5*V(K2)*V(K2))*AMX(K2)
      EO3 = (AIX(K3) + 0.5*V(K3)*V(K3))*AMX(K3)
      AMX(K2) = Z(94)
      AMX(K3) = AMX(K2)
      AIX(K2) = Z(93)
      AIX(K3) = Z(93)
      V(K2) = DETVEL
      V(K3) = DETVEL
      EN2 = (AIX(K2) + 0.5*V(K2)*V(K2))*AMX(K2)
      EN3 = (AIX(K3) + 0.5*V(K3)*V(K3))*AMX(K3)
  300 ETH = ETH + EN2 + EN3 - EO2 - EO3
      IF (NC - 2) 1,2,2
    1 ENSUM = 0.0
      GO TO 3000
    2 DETLEN = DETVEL*DT
      ENDEP = DETLEN*TMIC*EXEN*ZAMO
      ENSUM = ENSUM + ENDEP
      WRITE (6,9444) DETLEN,ENDEP,ENSUM
 9444 FORMAT (22HDETONATION LENGTH IS ,1PE12.3,10X,
     1  20HENERGY DEPOSITED IS ,1PE12.3,10X,
     2  26HTOTAL ENERGY DEPOSITED IS ,1PE12.3,/)
      JCELL = AJCEL
      KCELL = 2 + (JCELL - 1)*IMAX
      AIX(KCELL) = AIX(KCELL) + ENDEP/AMX(KCELL)
      ETH = ETH + ENDEP
 3000 VEL=0.0
 3005 DO 3050 I=1,IMAX
```

```
 3010 K=I+1
 3015 DO 3050 J=1,JMAX
 3020 IF(ABS(U(K))-9999.)3050,3050,3025
C
 3025 UMAX=1.0
      RMS=0.0
      P(K)=0.5*ABS(U(K)/TAU(I))/DT*U(K)
C
 3030 IF(ABSF(P(K))-1.0E-20)3035,3035,3040
 3035 P(K)=0.0
 3040 IF(UMAX-VEL)3050,3050,3045
 3045 VEL=UMAX
 3050 K=K+IMAX
 3055 KDT=1
      UMAX=-1.0
 3115 IT=0
 3120 IF(DT-SIG)3120,3170,3170
 3125 IT=IT
 3170 DO 3255 I=1,IMAX
 3175 K=I-1
 3180 DO 3255 J=1,JMAX
 3185 KP=K+IMAX
      IF(ABS(U(K))-9999.)3255,3200
 3200 RMS=SQRTF(G*ABS(TAU(I)*DT*SIN(P(K))/(ABS(U(K))))
 3205 RMS=RMS/SIG
 3210 IF(UMAX-RMS)3215,3,3
 3215 MIN=I
      MIN=J
      UMAX=RMS
    3 IF(GAMMA-1.)4,4,5
    4 RMS=ABSF(U(K))/TAU(I)*A(I))/0.5*PI*DT
      GO TO 3225
    5 RMS=ABSF(U(K))/DT
 3225 IF(UMAX-RMS)3250,3255,3255
 3230 UMAX=RMS
      MIN=I
      MIN=J
 3235 RMS=ABSF(V(K))/DT
 3240 IF(UMAX-RMS)3245,3250,3250
 3245 MIN=I
      MIN=J
      UMAX=RMS
 3250 CONTINUE
 3255 K=K+IMAX
      IF(UMAX-9999.)3260,3260,3260
 3260 DT=.5/VEL/UMAX*2(I-1)
 3295 KDT=0
 3300 T=T+.5*DT
C    THE FOLLOWING SET OF LOOPS DETERMINES A DT SUCH THAT THE HEAT CYCLE
C     WILL HAVE A STANDARD TIME IF THE CALCULATED DT WILL MAKE THE NEW
C     TIME GREATER THAN THE STANDARD
      CALL POWER(T,DEA,LP)
      DEA=(T+DT)/10.**LP
      IF(DEA.LE.2.)1201,202
  201 DO 210 LX=1,12
      EX=(LX-1.)/10.
      IF(DEA.LT.(2.+EX).AND.DEA.GE.(1.+EX))211,210
  210 CONTINUE
```

```
      GO TO 88
  207 IF(NCR.LE.2.)211,211,208
  208 DO 212 L=1,2
      E=(2L+2*L-1)/10.
      IF(NCR.LT.E+.001).AND.(NCR.GE.E-.001)211,212
  212 CONTINUE
      GO TO 88
  214 IF(NCR.LE.3.)215,215,216
  215 EA=2.5
      IF(NCR.LT.E+.001).AND.(NCR.GE.E-.001)211,88
  216 IF(NCR.LE.4.)217,217,218
  217 DO 219 L=1,2
      E=(2*2.+3*L-1)/10.
      IF(NCR.LT.E+.001).AND.(NCR.GE.E-.001)211,218
  218 CONTINUE
      GO TO 88
  219 IF(NCR.LE.5.)221,221,220
  220 E=5.
      IF(NCR.LT.E+.001).AND.(NCR.GE.E-.001)211,88
  221 IF(NCR.LE.6.)223,223,222
  222 DO 224 L=1,2
      E=(3*2.+3*L-1)/10.
      IF(NCR.LT.E+.001).AND.(NCR.GE.E-.001)211,222
  223 CONTINUE
      GO TO 88
  224 IF(NCR.LE.7.)226,226,225
  225 E=6.
      IF(NCR.LT.E+.001).AND.(NCR.GE.E-.001)211,88
  226 DO 227L=1,4
      E=(4*2.+3*L-1)/10.
      IF(NCR.LT.E+.001).AND.(NCR.GE.E-.001)211,227
  227 CONTINUE
      GO TO 88
  228 IF(ABS(F(E-NC(1)-1.E-9)).0),229
  229 DT=(E-NC(1)*10.**LP
      IF (NC - 100) 25 ,251,260
  250 PRINT = 5.0
      PRINTS = 5.0
      DUMPIT = 5.0
      GO TO 88
  260 PRINTL = 200.0
      PRINTS = 200.0
      DUMPIT = 200.0
C
   90 NC=NC+1
      CYCLE=NC
      NPC=NPC+1
 3305 IF(T)9999,3320,3310
 3310 IF(KDT)3310,3315,3320
 3315 WRITE (6,4000)T,DTNA,DT
 3320 DTNA=DT
      GO TO 3325
```

```
C                    NEGATIVE ERR
 9991 NX=3328
      GO TO 9995
 9992 NX=3335
      GO TO 9995
 9993 NX=3336
      GO TO 9995
C     THIS IF WILL BE 1, OR NEGATIVE ,STOP
 9994 NX=1
      GO TO 9995
 9995 NX=0
      NX=2
      CALL EXIT(NX,NX)
 1996 RETURN
 9996 FORMAT(1X,'NUMBER OF . . ',I=,I2.1,11H      JT(N)=,I2.3,13H      D
     1I(N)=,I2.1)
      END



      SUBROUTINE MAIN
      DIMENSION A(6(15000),AMA(3330),P(3519),
     1 U(23586),V(15000),R(30),Y(1100),Z(1150),
     2 IP(1100),IBJ(500),PL(1300),UL(1300),PP(1300),
     3 RR(1100),IWL(150),FLOFF(1100),YAMC(1100),
     4 SIGC(1000),GAMC(1100)
      DIMENSION BL(1100)
C
C                C     O     M     M     O     N
C
      COMMON          Z          ,XO        ,X         ,Y
      COMMON          XIX        ,A1        ,1AX       ,AREA      ,L
      COMMON          FS         ,T         ,II        ,JR        ,IMS
      COMMON          J          ,K         ,KP        ,L         ,M
      COMMON          MZ         ,N         ,MK        ,NO        ,NOP
      COMMON          NR         ,P         ,PAPOVL    ,PBLO      ,PIDTS
      COMMON          PRR        ,RC        ,REZ       ,RHO       ,RR
      COMMON          SIG        ,TAU       ,TAUDTS    ,TTX       ,TTY
      COMMON          U          ,URR       ,UT        ,UTEF      ,UU
      COMMON          UVMAX      ,V         ,VABOVE    ,VBLO      ,VEL
      COMMON          VTEF       ,WS        ,WSA       ,WSB       ,WSC
      COMMON          XLF        ,XR        ,YLW
      COMMON          YU         ,UR        ,PR
C
C                E   Q   U   I   V   A   L   E   N   C   E
C
      EQUIVALENCE     (Z,IZ,PROB),     (Z(2),CYCLE),     (Z(3),DT),
     1(Z(4),PRINTS),  (Z(5),PRINTL),   (Z(6),DUMPTT),    (Z(7),CSTOP),
     2(Z(8),PIDV),    (Z(9),TMZ),      (Z(10),GAM),      (Z(11),GAMD),
     3(Z(12),GAMX),   (Z(13),ETM),     (Z(14),FFA),      (Z(15),FFB),
     4(Z(16),TMDZ),   (Z(17),TMXZ),    (Z(18),XMAX),     (Z(19),TXMAX),
     5(Z(20),TYMAX),  (Z(21),AMDM),    (Z(22),AMXM),     (Z(23),DNN),
     6(Z(24),DMIN),   (Z(25),FEF),     (Z(26),DTMA),     (Z(27),CVIS),
     7(Z(28),NPRT),   (Z(29),NPRI),    (Z(30),NC),       (Z(31),NPC),
     8(Z(32),NRC),    (Z(33),IMAX),    (Z(34),IMAXA),    (Z(35),JMAX),
     9(Z(36),JMAXA),  (Z(37),KMAX),    (Z(38),KMAXA),    (Z(39),NMAX)
      EQUIVALENCE     (Z(40),ND),      (Z(41),KDT),      (Z(42),IXMAX),
     1(Z(43),NOD),    (Z(44),NOPR),    (Z(45),NIMAX),    (Z(46),NJMAX),
```

```
     2(Z(47),I1),      (Z(48),I2),      (Z(49),I3),      (Z(50),I4),
     3(Z(51),N1),      (Z(52),N2),      (Z(53),N3),      (Z(54),N4),
     4(Z(55),N5),      (Z(56),N6),      (Z(57),N7),      (Z(58),N8),
     5(Z(59),N9),      (Z(60),N10),     (Z(61),N11),     (Z(62),NRM),
     6(Z(63),TRAD),    (Z(64),XNRG),    (Z(65),SN),      (Z(66),DXN),
     7(Z(67),RADER),   (Z(68),PADET),   (Z(69),RADEB),   (Z(70),DTRAD),
     8(Z(71),REZFCT),  (Z(72),RSTOP),   (Z(73),SHELL),   (Z(74),BBOUND),
     9(Z(75),TOZONE),  (Z(76),ECK),     (Z(77),SBOUND),  (Z(78),X1)
     0EQUIVALENCE      (Z(79),X2),      (Z(80),Y1),      (Z(81),Y2),
     1(Z(82),CABLN),   (Z(83),VISC),    (Z(84),T),       (Z(85),GMAX),
     2(Z(86),WSGO),    (Z(87),WSGX),    (Z(88),GMADR),   (Z(89),GMAXP),
     3(Z(90),S1),      (Z(91),S2),      (Z(92),S3),      (Z(93),S4),
     4(Z(94),S5),      (Z(95),S6),      (Z(96),S7),      (Z(97),S8),
     5(Z(98),S9),      (Z(99),S10),
     6(Z(127),DX),     (Z(128),DY1),    (Z(129),DY),
     7(Z(130),NFITB),  (Z(131),NFITT),  (Z(132),NFITR),
     8(Z(133),NPARB),  (Z(134),NPART),  (Z(135),NPARR)
C
      EQUIVALENCE      (P,IW1)
      EQUIVALENCE(P(200),GAMC),(P(400),YAMC)
      EQUIVALENCE      (UR,UL,FLEFT),   (PR,PL,SIGC)
      EQUIVALENCE  (Z(120),DETVEL),  (Z(122),EXEN)
      EQUIVALENCE  (Z(123),EXPHO),   (Z(124),THICK)
C
C     ****** NOTE 1 MATERIAL ONLY (X)) ************
C
C
C
C     ==========================================
C
C
      UU=1.E+15
      UT=0.0
 8000 VEL=1.0
 8001 IF(GAM)9000,3301,9000
 9000 RC=1.0
      RR=RC
      GO TO 3304
 3301 RC=DX/2.0
      RR=(X(1)+X(2))/2.0
 3304 K=2
      DO 3302 J=1,JMAX
      PL(J)=P(K)
      UL(J)=0.0
      QL(J)=0.0
 3302 K=K+IMAX
      DO 3360 I=1,IMAX
      K=I+1
    6 NOT=IMAX
    8 K=I+1+NOT
      IF(CVIS)7002,7003,7003
 7003 VBLO=0.0
      PBLO=P(K)
      GO TO 7004
 7002 VBLO=V(K)
      PBLO=0.0
 7004 TAUDTS=TAU(I)*DT
      QB=0.0
```

```
    3 NO=2
    4 DO 3348 J=NO,JMAX
      UQ=U(K)
      VQ=V(K)
      A=0.
      B=0.
      PIDTS=1.0/(PIDY*DT*DY)
      IF(GAM)9002,9004,9002
 9002 PIDTS=2.0*PIDTS
 9004 N=K+IMAX
 3305 IF(AMX(K))9902,3340,3306
 3306 IF(IMAX-I)9903,3311,3310
 3310 IF(AMX(K+1))9904,3312,3314
 3311 PRR=P(K)
      QR=0.0
 3307 ETH=ETH-PRR*U(K)/PIDTS*RC
      GO TO 3313
 3312 PRR=0.0
      QR=0.0
 3313 URR=RC*U(K)
      GO TO 3315
 3314 PRR=(P(K)+P(K+1))/2.0
 3315 URR=(U(K)*RC+U(K+1)*RR)/2.0
      RHO=0.5*(AMX(K)/TAU(I)+AMX(K+1)/TAU(I+1))/DY
      DUR=U(K)-U(K+1)
      IF(DUR)2003,10001,2005
 2003 IF(S3)2004,10001,2004
10001 QR=0.0
      GO TO 3316
 2004 UTEF=ABSF(0.5*(U(K)+U(K+1)))
      QR=-S3*RHO*UTEF*DUR
      GO TO 3316
 2005 IF(S3)2007,2006,2007
 2006 UTEF=0.
      GO TO 2008
 2007 UTEF=ABSF(0.5*(U(K)+U(K+1)))
 2008 CS=SQRTF(1.4*PRR/RHO)
      QR=RHO*DUR*(S1*CS+S2*S2*DUR+S3*UTEF)
 3316 IF(JMAX-J)9905,3318,3320
 3318 PABOVE=P(K)
      QT=0.0
      A=1.
 3319 ETH=ETH-PABOVE*V(K)/2.0*TAUDTS
      GO TO 3323
 3320 IF(AMX(N))9906,3322,3324
 3322 PABOVE=0.0
      QT=0.0
 3323 VABOVE=V(K)
      GO TO 3328
 3324 PABOVE=(P(K)+P(N))/2.0
      RHO=0.5*(AMX(K)+AMX(N))/TAJ(I)/DY
      DVZ=V(K)-V(N)
      IF(DVZ)2009,10003,2011
 2009 IF(S3)2010,10003,2010
10003 QT=0.0
      GO TO 7001
 2010 VTEF=ABSF(0.5*(V(K)+V(N)))
      QT=-S3*RHO*VTEF*DVZ
```

```
      GO TO 7001
2011  IF(S3)2016,2012,2016
2012  VTEF=0.
      GO TO 2014
2016  VTEF=ABSF(0.5*(V(K)+V(N)))
2014  CS=SQRTF(1.4*PABOVE/RHO)
      QT=RHO*DVZ*(S1*CS+S2*S2*DVZ+S3*VTEF)
7001  IF(NO-J) 3325,7005,9905
7005  IF(CVIS) 7000,7006,7006
7000  PBLO=P(K)
      ETH=ETH+PBLO*V(K)/2.0*TAUDTS
7006  B=1
3325  VABOVE=(V(K)+V(N))/2.0
3328  IF(VEL) 9907,6016,3400
3400  G=6.3567E+08
      G=Z(108)*G*G/((G+Y(J)-DY/2.)**2)
      IF(A+B) 9,9,10
    9 WS=(PBLO-PABOVE+QB-QT)*TAUDTS/AMX(K)-G*DT
      GO TO 11
   10 WS=(PBLO-PABOVE+QB-QT)*TAUDTS/AMX(K)-0.5*G*DT
   11 V(K)=V(K)+WS
      IF(ABSF(V(K))-1.E-01) 3401,3401,3402
3401  V(K)=0.0
3402  U(K)=U(K)+(PL(J)-PRR+QL(J)-QR)/(AMX(K))*RC/PIDTS*2.0
      IF(ABSF(U(K))-1.E-01) 3403,3403,6016
3403  U(K)=0.0
6016  CONTINUE
      IF(VEL)9907,2001,2000
2000  VQ=(VQ+V(K))/2.0
      UQ=(UQ+U(K))/2.0
      WS=(.5*P(K)*(VBLO-VABOVE)+QB*(VBLO-VQ)+QT*(VQ-VABOVE))*TAUDTS
      WT=(.5*P(K)*(UL(J)-URR)+QL(J)*(UL(J)-RC*UQ)+QR*(RC*UQ-URR))/PIDTS*
     12.0
      GO TO 2002
2001  WS=P(K)*(VBLO-VABOVE)*TAUDTS/2.0
      WT=P(K)*(UL(J)-URR)/PIDTS
2002  RHO=WS+WT
3332  WSX=AIX(K)+RHO/AMX(K)
1000  IF(WSX)1011,1001,1001
1001  AIX(K)=WSX
      GO TO 3342
1011  UT=1.0
      PRINT 2013
2013  FORMAT(4X,1HI,3X,1HJ,5X,3HWSX,10X,3HAIX,10X,3HAMX,10X,3H  P,9X,2HW
     1T,11X,2HWS,11X,2HDT,11X,3HVEL)
      PRINT 2015,I,J,WSX,AIX(K),AMX(K),P(K),WT,WS,DT,VEL
2015  FORMAT(X,2I4,X,7(E12.4,X),F4.1,//)
      WSA=2.0*AIX(K)/3.0*DT/(AIX(K)-WSX)
1013  IF(WSA-UU)1014,1001,1001
1014  UU=WSA
      ISAVE=I
      JSAVE=J
      GO TO 1001
3340  PRR=0.0
      QR=0.0
      QT=0.0
      URR=U(K+1)*RR
      PABOVE=0.0
```

```
      VABOVE=V(N)
 3342 VBLO=VABOVE
      PL(J)=PRR
      UL(J)=URR
      QL(J)=QR
      QB=QT
      K=N
 3348 PBLO=PABOVE
 3355 RC=RR
      IF(GAM)3360,9007,3360
 9007 RR=(X(I+1)+X(I+2))/2.0
 3360 CONTINUE
 3361 IF(VEL)9911,10000,3363
 3363 VEL=0.0
      GO TO 8001
C                 ERROR
 9902 NK=3305
      GO TO 9999
 9903 NK=3306
      GO TO 9999
 9904 NK=3310
      GO TO 9999
 9905 NK=3316
      GO TO 9999
 9906 NK=3320
      GO TO 9999
 9907 NK=3328
      GO TO 9999
 9911 NK=3361
 9999 NR=3
      CALL ERRER(NR,NK)
10000 IF(SN)6201,7031,6201
 7031 IF(UT)7020,6201,7010
 7010 UT=-1.0
      DT=-DT
      GO TO 8000
 7020 UT=0.0
      PRINT 2060,ISAVE,JSAVE,UU
 2060 FORMAT(8H CELL I=,I3,3H J=,I3,25H FINALLY SETS TIME STEP =,E12.4)
      DT=UU
      DTNA=DT
      GO TO 8000
 6201 RETURN
      END




      SUBROUTINE EDIT
C
C                 D   I   M   E   N   S   I   O   N
C
      DIMENSION AIX(3500),AMX(3500),P(3500),
     1 U(3500),V(3500),X(60),Y(100),Z(150),
     2 IZ(150),TAU(60),PL(100),UL(100),PR(100),
     3 UR(100),IW1(15),FLEFT(100),YAMC(100),
     4 SIGC(100),GAMC(100)
```

```
C
C
C                C     O     M     M     O     N
C
C
C
      COMMON        Z         ,XO       ,X        ,Y
      COMMON        AIX       ,AM       ,AMX      ,AREA     ,E
      COMMON        FS        ,I        ,II       ,IR       ,IWS
      COMMON        J         ,K        ,KP       ,L        ,M
      COMMON        MZ        ,N        ,NK       ,NO       ,NOP
      COMMON        NR        ,P        ,PABOVE   ,PBLO     ,PIDTS
      COMMON        PRR       ,RC       ,REZ      ,RHO      ,RR
      COMMON        SIG       ,TAU      ,TAUDTS   ,TTX      ,TTY
      COMMON        U         ,URR      ,UT       ,UTEF     ,UU
      COMMON        UVMAX     ,V        ,VABOVE   ,VBLO     ,VEL
      COMMON        VTEF      ,WS       ,WSA      ,WSB      ,WSC
      COMMON        XLF       ,XR       ,YLW
      COMMON        YU        ,UR       ,PR
C
C
C                E   Q   U   I   V   A   L   E   N   C   E
C
     1EQUIVALENCE       (Z,IZ,PROB),      (Z(2),CYCLE),     (Z(3),DT),
     1(Z(4),PRINTS),    (Z(5),PRINTL),    (Z(6),DUMPT7),    (Z(7),CSTOP),
     2(Z(8),PIDY),      (Z(9),TMZ),       (Z(10),GAM),      (Z(11),GAMO),
     3(Z(12),GAMX),     (Z(13),ETH),      (Z(14),FFA),      (Z(15),FFB),
     4(Z(16),TMDZ),     (Z(17),TMXZ),     (Z(18),XMAX),     (Z(19),TXMAX),
     5(Z(20),TYMAX),    (Z(21),AMDM),     (Z(22),AMXM),     (Z(23),DMIN),
     6(Z(24),DMIN),     (Z(25),FEF),      (Z(26),DTNA),     (Z(27),CVIS),
     7(Z(28),NPR),      (Z(29),NPRI),     (Z(30),NC),       (Z(31),NPC),
     8(Z(32),NRC),      (Z(33),IMAX),     (Z(34),IMAXA),    (Z(35),JMAX),
     9(Z(36),JMAXA),    (Z(37),KMAX),     (Z(38),KMAXA),    (Z(39),NMAX)
     0EQUIVALENCE       (Z(40),ND),       (Z(41),KDT),      (Z(42),IXMAX),
     1(Z(43),NOD),      (Z(44),NOPP),     (Z(45),NIMAX),    (Z(46),NJMAX),
     2(Z(47),I1),       (Z(48),I2),       (Z(49),I3),       (Z(50),I4),
     3(Z(51),N1),       (Z(52),N2),       (Z(53),N3),       (Z(54),N4),
     4(Z(55),N5),       (Z(56),N6),       (Z(57),N7),       (Z(58),N8),
     5(Z(59),N9),       (Z(60),N10),      (Z(61),N11),      (Z(62),NRM),
     6(Z(63),TRAD),     (Z(64),XNRG),     (Z(65),SN),       (Z(66),DXN),
     7(Z(67),RADER),    (Z(68),RADET),    (Z(69),RADEB),    (Z(70),DTRAD),
     8(Z(71),REZFCT),   (Z(72),PSTOP),    (Z(73),SHELL),    (Z(74),BBOUND),
     9(Z(75),TOZONE),   (Z(76),ECX),      (Z(77),SBOUND),   (Z(78),X1)
     0EQUIVALENCE       (Z(79),X2),       (Z(80),Y1),       (Z(81),Y2),
     1(Z(82),CABLN),    (Z(83),VISC),     (Z(84),T),        (Z(85),GMAX),
     2(Z(86),WSGD),     (Z(87),WSGX),     (Z(88),GMADR),    (Z(89),GMAXR),
     3(Z(90),S1),       (Z(91),S2),       (Z(92),S3),       (Z(93),S4),
     4(Z(94),S5),       (Z(95),S6),       (Z(96),S7),       (Z(97),S8),
     5(Z(98),S9),       (Z(99),S10),
     6(Z(127),DX),      (Z(128),DY1),     (Z(129),DY),
     7(Z(130),NFITB),   (Z(131),NFITT),   (Z(132),NFITR),
     8(Z(133),NPARB),   (Z(134),NPART),   (Z(135),NPARR)
C
      EQUIVALENCE       (P,IW1)
      EQUIVALENCE(P(200),GAMC),(P(400),YAMC)
      EQUIVALENCE       (UR,UL,FLEFT),    (PR,PL,SIGC)
C
      EQUIVALENCE   (Z(120),DETVEL),   (Z(122),EXEN)
      EQUIVALENCE   (Z(123),EXRHO),    (Z(124),THICK)
```

```
C
C
  102 IF(SENSE SWITCH 4)122,104
  104 IF(SENSE LIGHT 3)106,108
  108 IF (CYCLE - CSTOP) 112,122,122
  112 IF(MODF(CYCLE,DUMPT7))114,1,114
  114 IF(MODF(CYCLE,PRINTL))120,126,120
  120 IF(MODF(CYCLE,PRINTS))140,6000,140
  140 IF(SENSE LIGHT 1)142,144
  142 REWIND 4
      IF (GAMD) 9,9,10
   10 REWIND 7
    9 SENSE LIGHT 1
      IF (GAMX) 144,144,143
  143 CALL STEADY
  144 GO TO 10000
C
C                DUMP ON TAPE 7
C
  122 SENSE LIGHT 1
      PRINTL=CYCLE
    1 IF (DJMPT7) 30,2,2
    2 IF (GAMD) 3,3,203
  203 BACKSPACE 7
  204 WS = 555.0
      WRITE (7) WS,CYCLE,N3
  220 WRITE (7) (Z(L),L=1,MZ)
  206 WRITE ( 7)(U(K),V(K),AMX(K),AIX(K),P(K),K=1,KMAXA)
  207 WRITE ( 7)X(0),(X(K),TAU(K),K=1,IMAX)
  221 WRITE ( 7)(Y(K-1),K=1,JMAXA)
  222 WS = 666.0
      WRITE  (7) WS,WS,WS
      GO TO 8
    3 BACKSPACE  4
C     ************************
C     NOTE,ADDITIONAL DATA ON TAPE 7
C     ************************
    4 WS=555.0
      WRITE ( 4)WS,CYCLE,N3
   20 WRITE ( 4)(Z(L),L=1,MZ)
    6 WRITE ( 4)(U(K),V(K),AMX(K),AIX(K),P(K),K=1,KMAXA)
    7 WRITE ( 4)X(0),(X(K),TAU(K),K=1,IMAX)
   21 WRITE ( 4)(Y(K-1),K=1,JMAXA)
   22 WS=666.
      WRITE ( 4)WS,WS,WS
    8 IF (SENSE SWITCH 2) 5,24
    5 REWIND 4
      PRINT 8121,CYCLE
 8121 FORMAT(7H CYCLE ,F5.0,18H BEGINS NEW TAPE 4)
      PAUSE 10
      GO TO 4
   24 WRITE (6,8120)NC
   30 GO TO 114
C
C
  106 SENSE LIGHT 3
  126 SENSE LIGHT 4
      IF (SENSE SWITCH 1) 11,6000
```

```
   11 CALL DISPLA (7HCYCLE = ,NC)
 6000 DO 6012 I=1,12
 6012 PR(I)=0.0
      DO 6028 K=2,KMAX
      WSB=(U(K)**2+V(K)**2)/2.0
 6017 PR(1)=0.0
      PR(2)=0.0
      PR(4)=0.0
 6019 IF(AMX(K))9917,6028,6020
 6020 PR(5)=PR(5)+AIX(K)*AMX(K)
      PR(6)=PR(6)+WSB*AMX(K)
      PR(8)=PR(8)+AMX(K)
 6028 CONTINUE
      PR(3)=PR(1)+PR(2)
      PR(7)=PR(5)+PR(6)
      XNRG=PR(7)
      PR(9)=PR(1)+PR(5)
      PR(10)=PR(2)+PR(6)
      PR(11)=PR(3)+PR(7)
      PR(12)=PR(4)+PR(8)
      WSA=(ETH-PR(11))/ETH
      PR(18)=(WSA-DNN)/FLOATF(NPC)
      ECK=PR(18)
      DNN=WSA
      NPC=0
      WRITE (6,8116)PROB,NC,T,DTNA,TRAD,DTRAD,NR,N1,N2,N3,N4
      WRITE (6,8117)(PR(I),I=1,8)
      WRITE (6,8118)(PR(I),I=9,12)
      WRITE (6,8119)UVMAX,ETH,ECK
      WRITE (6,9040)N10,N11,I1,I2,I3,I4
 6090 IF(SENSE LIGHT 4)5000,136
C**** END OF   S P  SUBROUTINE ***************************************
C
C
C**** SUBROUTINE L P *************************************************
5000  CONTINUE
 5001 WRITE (6,8116)PROB,NC,T,DTNA,TRAD,DTRAD,NR,N1,N2,N3,N4
 5004 DO 5050 I=1,IMAX
      SENSE LIGHT 4
      J=JMAX+1
      K=JMAX*IMAX+I+1
      DO 5046 L=1,JMAX
      J=J-1
      K=K-IMAX
 5012 IF(AMX(K))9917,5046,5014
 5014 IF(SENSE LIGHT 4)5016,5019
 5016 WRITE (6,8135)I,X(I),DX
 5019 WS=AMX(K)/(TAU(I)*DY)
 5018 WRITE(6,8108)J,U(K),V(K),P(K),AMX(K),WS,AIX(K),Y(J)
 5046 CONTINUE
      IF (CYCLE - CSTOP) 136,5050,136
 5050 CONTINUE
  136 IF (ABSF(ECK)-DMIN)140,140,9905
C**** END OF   L P  SUBROUTINE ***************************************
C
C
C                ERROR
 9901 NK=110
```

```
      GO TO 9999
C                 ENERGY CHECK
 9905 NK=136
      GO TO 9999
C                 NEGATIVE MASS
 9917 NK=6015
      GO TO 9999
 9920 NK=904
      GO TO 9999
 9921 NK=912
      GO TO 9999
 9922 NK=918
      GO TO 9999
 9923 NK=922
      GO TO 9999
 9924 NK=926
 9999 NR=6
      CALL ERRER(NR,NK)
10000 RETURN
C
C                 FORMATS
 8108 FORMAT(I4,1X,1P7E16.6)
 81160FORMAT(8H1PROBLEM6X,5HCYCLE9X,4HTIME13X,2HDT13X,4HTRAD11X,5HDTRAD1
     12X,2HNR6X,2HN14X,2HN24X,2HN34X,2HN4/(F9.3,I10,1X,1P4E16.7,I10,2X,4
     2I6))
 81170FORMAT(1H0//17X2HAI16X,2HAK14X,5HAI+AK15X,2HAM/4H DOT3X,1P4E18.7/3
     1H  X4X,4E18.7)
 81180FORMAT(12X,13H-------------5X,13H-------------5X,13H-------------5
     1X,13H-------------/7H TOTALS1P4E18.7)
 81190FORMAT(2H0 //16X,7HMAX VEL13X,3HTHE12X,9HREL ERROR/6X,1P6E19.7///)
 8120 FORMAT(1H0//21H TAPE 7 DUMP ON CYCLEI5////)
 81350FORMAT(1H ///4H I =I3,6X,6HX(I) =F12.3,3X,6H  DX =F12.3//3H  J10X,
     11HX15X,1HY14X,3HF/A13X,3HAMX13X,3HRHO13X,3HAIX14X,1HY/)
 9040 FORMAT(1H / 6I6)
      END




      SUBROUTINE POWER(X,XX,N)
      XX=LOGF(ABSF(X))/2.302585093
      N=XX
      IF(XX)1,2,2
    1 N=N-1
    2 XX=X/10.**N
      RETURN
      END



      SUBROUTINE PH2
C
CC                D   I   M   E   N   S   I   O   N
C
      DIMENSION AIX(3500),AMX(3500),P(3500),
     1 U(3500),V(3500),X(60),Y(100),Z(150),
```

```
     2 IZ(150),TAU(60),PL(100),UL(100),PR(100),
     3 UR(100),IW1(15),FLEFT(100),VAMC(100),
     4 SIGC(100),GAMC(100)
C
C                C     O     M     M     O     N
C
      COMMON         Z         ,XO       ,X        ,Y
      COMMON         AIR       ,AM       ,AMR      ,AREA      ,C
      COMMON         FS        ,I        ,II       ,IR        ,IWS
      COMMON         J         ,K        ,KP       ,L         ,M
      COMMON         MZ        ,N        ,NK       ,NO        ,NOP
      COMMON         NR        ,P        ,PABOVE   ,PBLO      ,PIDTS
      COMMON         PRR       ,RC       ,REZ      ,RMO       ,RR
      COMMON         SIG       ,TAU      ,TAUDTS   ,TTR       ,TTV
      COMMON         U         ,URR      ,UT       ,UTEF      ,UU
      COMMON         UMAX      ,V        ,VABOVE   ,VBLO      ,VLL
      COMMON         VTEF      ,WS       ,WSA      ,WSB       ,WSC
      COMMON         XLF       ,XR       ,YLM
      COMMON         YU        ,UR       ,PR
C
C                E   Q   U   I   V   A   L   E   N   C   E
C
     0EQUIVALENCE       (Z,IZ,PABO),      (Z(2),CYCLE),     (Z(3),DT),
     1(Z(4),PRINTS),    (Z(5),PRINTL),    (Z(6),DUMPIT),    (Z(7),CSTOP),
     2(Z(8),PIDY),      (Z(9),IMZ),       (Z(10),GA4),      (Z(11),GAMO),
     3(Z(12),GAMK),     (Z(13),E14),      (Z(14),FFA),      (Z(15),FFM),
     4(Z(16),IMO2),     (Z(17),IMG2),     (Z(18),XMAX),     (Z(19),IXMAX),
     5(Z(20),IYMAX),    (Z(21),AMOM),     (Z(22),AMOM),     (Z(23),DVM),
     6(Z(24),DMIN),     (Z(25),FEF),      (Z(26),DTAA),     (Z(27),CVIS),
     7(Z(28),NP2),      (Z(29),NP2I),     (Z(30),NC),       (Z(31),NPC),
     8(Z(32),NRC),      (Z(33),IMAX),     (Z(34),IMAXA),    (Z(35),JMAX),
     9(Z(36),JMAXA),    (Z(37),KMAX),     (Z(38),KMAXA),    (Z(39),NMAX)
     0EQUIVALENCE       (Z(40),NJ),       (Z(41),KDT),      (Z(42),LMAX),
     1(Z(43),NOT),      (Z(44),NOPR),     (Z(45),NIMAX),    (Z(46),NJMAX),
     2(Z(47),I1),       (Z(48),I2),       (Z(49),I3),       (Z(50),I4),
     3(Z(51),N1),       (Z(52),N2),       (Z(53),N3),       (Z(54),N4),
     4(Z(55),N5),       (Z(56),N6),       (Z(57),N7),       (Z(58),N8),
     5(Z(59),N9),       (Z(60),N10),      (Z(61),N11),      (Z(62),N2M),
     6(Z(63),TRAD),     (Z(64),XNRG),     (Z(65),SM),       (Z(66),DEN),
     7(Z(67),RADER),    (Z(68),RADLT),    (Z(69),RADEN),    (Z(70),DTRAD),
     8(Z(71),PEZECT),   (Z(72),PSTOP),    (Z(73),SMELL),    (Z(74),NBOUND),
     9(Z(75),TOZONE),   (Z(76),FCK),      (Z(77),SBOUND),   (Z(78),X1)
     0EQUIVALENCE       (Z(79),X2),       (Z(80),Y1),       (Z(81),Y2),
     1(Z(82),CABLN),    (Z(83),VISC),     (Z(84),T),        (Z(85),GMAX),
     2(Z(86),WSG),      (Z(87),WSGX),     (Z(88),GMADR),    (Z(89),GMAXR),
     3(Z(90),S1),       (Z(91),S2),       (Z(92),S3),       (Z(93),S4),
     4(Z(94),S5),       (Z(95),S6),       (Z(96),S7),       (Z(97),S8),
     5(Z(98),S9),       (Z(99),S10),
     6(Z(127),DX),      (Z(128),DY1),     (Z(129),DY),
     7(Z(130),NFITB),   (Z(131),NFITT),   (Z(132),NFITR),
     8(Z(133),NPAPB),   (Z(134),NPAPT),   (Z(135),NPAPR)
C
      EQUIVALENCE       (P,IW1)
      EQUIVALENCE(P(200),GAMC),(P(400),VAMC)
      EQUIVALENCE       (UR,UL,FLEFT),    (PR,PL,SIGC)
      EQUIVALENCE   (Z(120),DETVEL),   (Z(122),EXLN)
      EQUIVALENCE   (Z(123),EXPHO),    (Z(124),THICK)
```

```
C
C
C
C
      JBOT=Z(1146)
      Z(1141)=0.
      Z(1142)=0.
      Z(1143)=0.
      QCZ=0.0
      SENSE LIGHT 0
      PIOTS=1.0/(PIOV*DT)
  101 DO 103 J=1,JMAX
  102 GAMC(J)=0.0
      FLEFT(J)=0.0
      VAMC(J)=0.0
  103 VIC(J)=0.0
  104 DO 547 I=1,IMAX
      J=2
  105 K=(J-1)*IMAX+I
   40 IF(AMC(K))40,40,41
   41 IF(V(K))42,47,47
   47 AMMV=0.0
      AMMV=0.
      AMMU=0.
      DELEQ=0.
      GO TO 107
   42 AMMV=AMC(K)*V(K)*DT/DY
      MS=AMC(K)
      IF(GAMC(J))7000,43,43
 7000 MS=AMC(K)*GAMC(J)
   43 IF(AMMV+MS)44,45,45
   44 AMMV=-MS
   45 IF(CVIS)106,106,105
  106 AMMU=AMMV*U(K)
      AMMV=AMMV*V(K)
      DELEQ=AIR(K)*(U(K)**2+V(K)**2)/2.0
      MS=(U(K)**2+V(K)**2)/2.
      ETA=ETA+AMMV*(AIR(K)+MS)
      IF(-AMMV/(TAU(I)*DY)-TOZONE)107,107,99
   99 QCZ=1.
      Z(1143)=1.
C     MASS HAS LEFT THE BOTTOM
      GO TO 107
  108 AMMV=0.0
      AMMU=0.0
      AMMV=0.0
      DELEQ=0.0
  107 DO 545 J=2,JMAX
  109 L=K+IMAX
      VEL=0.0
      FS=0.0
  210 IF(JMAX-J)211,211,217
  211 VEL=1.0
      GO TO 204
  217 IF(AMC(L))213,213,214
  214 IF(AMC(K))216,216,239
  216 VAMOVE=V(L)
```

```
      GO TO 212
  215 IF(AMX(K))205,235,204
  205 VABOVE=0.0
      GO TO 212
  208 VABOVE=V(K)
      GO TO 212
  209 VABOVE=(V(K)+V(L))/2.0
  212 CONTINUE
      FS=0.0
  404 IF(IMAX-I)412,412,405
  405 IF(AMX(K+1))411,411,409
  409 IF(AMX(K))410,410,407
  410 URR=U(K+1)
      GO TO 408
  411 IF(AMX(K))403,403,406
  403 URR=0.0
      GO TO 408
  412 FS=1.0
  406 URR=U(K)
      GO TO 408
  407 URR=(U(K)+U(K+1))/2.0
  408 CONTINUE
  301 IF(VABOVE)300,304,302
  302 IF(AMX(K))9901,304,4800
 9900 IF(J-1)9901,303,4401
 4401 KP=K-IMAX
      IF(AMX(KP))9900,303,303
  303 M=K
      JJ=J
      GO TO 307
  304 AMPY=0.0
  304 AMUT=0.0
      AMVT=0.0
      DELET=0.0
      GO TO 501
  300 IF(VEL)9901,305,304
  305 IF(AMX(L))9903,304,306
  306 M=L
      JJ=J+1
  307 IF(VEL)6130,6130,6140
 6130 WSA=(V(K)+V(L))/2.0
      WSB=1.0+(V(L)-V(K))/(DY*SOUND)*DT
      VABOVE=WSA/WSB
 6140 AMPY=AMX(M)*VABOVE/DY*DT
  501 IF(URR)500,504,502
  502 IF(AMX(K))9900,504,503
  503 M=K
      N=I
      GO TO 508
  504 AMMP=0.0
      AMUR=0.0
      AMVR=0.0
      DELER=0.0
      GO TO 1
  508 IF(FS)9905,506,504
  506 IF(AMX(K+1))9904,504,507
  507 M=K+1
```

```
      N=I+1
  508 IF(FS)6100,6100,6110
 6100 WSA=(J(K)+U(K+1))/2.0
      WSB=1.0+(U(K+1)-U(K))/(DX*SBOUND)*DT
      URR=WSA/WSB
 6110 DEN=AMX(M)/TAU(N)
      IF(GAM)9990,9989,9990
 9989 CART=X(I)*2.0
      GO TO 9991
 9990 CART=1.0
 9991 AMMP=DEN/PIDTS*CART*URR
    1 IF(AMMP)16,74,8820
 8820 IF(GAMC(J))74,74,8821
 8821 IF(FS)6120,6120,74
 6120 IF(AMX(K+1))9903,74,74
   74 JTAG=0
    2 IF(AMPY)3,4,4
    3 ITAG=1
      WSB=AMPY
      AMPY=0.0
      GO TO 64
    4 ITAG=0
   64 IF(AMMY)9,5,5
    5 IF(GAMC(J))7,6,6
    6 WS=AMX(K)
      GO TO 11
    7 WS=AMX(K)+GAMC(J)
      GO TO 11
    9 IF(GAMC(J))10,8,8
    8 WS=AMX(K)+AMMY
      GO TO 11
   10 WS=AMX(K)+GAMC(J)+AMMY
   11 WSA=AMPY+AMMP
   12 IF(WSA-WS)75,75,13
   13 AMPY=AMPY*WS/WSA
      AMMP=AMMP*WS/WSA
   75 IF(JTAG)14,73,14
   73 WSC=AMMP
   14 IF(ITAG)15,7000,15
   15 AMPY=WSB
      ITAG=0
      GO TO 40
   16 IF(FS)76,17,76
   76 WSC=AMMP
      GO TO 40
   17 IF(I+1-IMAX)18,18,9908
   18 URRR=J(K+1)/2.0
      GO TO 20
   19 URRR=(U(K+1)+U(K+2))/2.0
   20 IF(URRR)39,39,21
   21 IF(GAM)9993,9992,9993
 9992 CART=X(I+1)*2.0
      GO TO 9994
 9993 CART=1.0
 9994 URRR=JRRR/TAU(I+1)*AMX(K+1)/PIDTS*CART
   22 IF(J-1)9909,23,24
   23 VBLO=V(K+1)/2.0
```

```
      GO TO 26
   24 KP=K+1-IMAX
      VBLO=(V(K+1)+V(KP))/2.0
   26 IF(VBLO)25,38,38
   25 VBLO=AMX(K+1)/DY*VBLO*DT
   27 IF(VEL)28,29,28
   28 VAB=V(K+1)/2.0
      GO TO 31
   29 KP=K+IMAX+1
      VAB=(V(K+1)+V(KP))/2.0
   31 IF(VAB)36,36,30
   30 VAB=AMX(K+1)/DY*VAB*DT
   32 WS=AMX(K+1)
   33 WSA=URRR-AMMP-VBLO+VAB
   34 IF(WSA-WS)77,77,35
   35 AMMP=AMMP*WS/WSA
   77 JTAG=1
      WSC=AMMP
      AMMP=0.0
      GO TO 2
   36 WS=AMX(K+1)
   37 WSA=URRR-AMMP-VBLO
      GO TO 34
   38 VBLO=0.0
      GO TO 27
   39 URRR=0.0
      GO TO 22
   40 IF(VEL)7000,41,7000
   41 IF(FS)42,43,42
   42 KP=K+IMAX
      URT=U(KP)/2.0
      GO TO 45
   43 KP=K+IMAX
      URT=(U(KP)+U(KP+1))/2.0
   45 IF(URT)46,46,70
   46 URT=0.0
      GO TO 47
   70 KP=K+IMAX
      IF(GAM)9997,9996,9997
 9996 CART=X(I)*2.0
      GO TO 9998
 9997 CART=1.0
 9998 URT=URT/TAU(I)*AMX(KP)/PIDTS*CART
   47 IF(J+1-JMAX)48,48,9910
   48 KP=K+IMAX
      KL=KP+IMAX
      VABT=(V(KP)+V(KL))/2.0
      GO TO 51
   49 KP=K+IMAX
      KL=KP+IMAX
      VABT=V(KP)/2.0
   51 IF(VABT)8810,71,72
 8810 IF(AMX(K))9903,71,71
   71 VABT=0.0
      GO TO 60
   72 VABT=VABT*AMX(KP)/DY*DT
   52 IF(GAMC(J+1))54,53,53
```

```
   53 WS=AMX(KP)
      GO TO 55
   54 WS=AMX(KP)+GAMC(J+1)
   55 WSA=VABT-AMPY+URT
      GO TO 57
   60 IF(GAMC(J+1))61,61,59
   61 WS=AMX(KP)+GAMC(J+1)
      GO TO 58
   59 WS=AMX(KP)
   58 WSA=-AMPY+URT
   57 IF(WSA-WS)7000,7000,56
   56 AMPY=AMPY*WS/WSA
      GO TO 7000
 7000 AMMP=WSC
  309 IF(AMPY)8834,8831,8833
 8833 IF(JMAX-J)9911,318,8835
 8835 KP=K+IMAX
 8836 IF(AMX(KP))9900,8837,318
 8837 IF(AMPY/(TAU(I)*DY    )-TOZONE)8838,318,318
 8838 AMPY=0.
      GO TO 8831
 8834 IF(J-1)9911,8831,8839
 8839 IF(AMX(K))9900,8840,5831
 8840 IF(-AMPY/(TAU(I)*DY    )-TOZONE)8841,8831,8831
 8841 AMPY=0.
      GO TO 8831
  318 DELM=GAMC(J)+AMMY-AMPY
  322 IF(VEL)9901,324,323
  323 WS=U(K)**2+V(K)**2
      ETH=ETH-AMPY*(AIX(K)+WS/2.0)
      IF(AMPY/(TAU(I)*DY)-TOZONE)324,324,6900
 6900 REZ=1.0
      Z(141)=1.
C     MASS HAS LEFT THE TOP
  324 AMUT=AMPY*U(K)
      AMVT=AMPY*V(K)
      GO TO 326
 8831 AMUT=AMPY*U(L)
      AMVT=AMPY*V(L)
      DELM=GAMC(J)-AMPY+AMMY
  326 IF(AMPY)327,328,328
  327 DELET=AIX(L)+(U(L)**2+V(L)**2)/2.0
      GO TO 333
  328 IF(AMMY)329,330,330
  329 DELET=DELEB
      GO TO 333
  330 IF(GAMC(J))331,332,332
  331 DELET=SIGC(J)
      GO TO 333
  332 DELET=AIX(K)+(U(K)**2+V(K)**2)/2.0
  333 SIGMU=FLEFT(J)+AMMU-AMUT
      SIGMV=YAMC(J)+AMMV-AMVT
      DELEK=GAMC(J)*SIGC(J)+AMMY*DELEB-AMPY*DELET
  509 IF(AMMP)8843,518,8844
 8844 IF(IMAX-I)9911,518,8845
 8845 IF(AMX(K+1))9900,8846,518
 8846 IF(AMMP/(TAU(I+1)*DY )-TOZONE)8847,518,518
```

```
 8847 AMMP=0.
      GO TO 518
 8843 IF(I-1)9911,512,8848
 8848 IF( AMX(K))9900,8849,512
 8849 IF(-AMMP/(TAU(I)*DY    )-TOZONE)8850,512,512
 8850 AMMP=0.
      GO TO 518
  512 DELM=DELM-AMMP+AMX(K)
 8828 AMUR=AMMP*U(K+1)
      AMVR=AMMP*V(K+1)
      GO TO 525
  518 DELM=DELM-AMMP+AMX(K)
  522 IF(FS)9905,524,523
  523 WS=U(K)**2+V(K)**2
      ETH=ETH-AMMP*(AIX(K)+WS/2.0)
      IF(AMMP/(TAU(I)*DY)-TOZONE)524,524,6901
 6901 REZ=1.
      Z(142)=1.
C     MASS HAS LEFT THE RIGHT SIDE
  524 AMUR=AMMP*U(K)
      AMVR=AMMP*V(K)
  525 SIGMU=SIGMU-AMUR
      SIGMV=SIGMV-AMVR
  526 TIC=0.0
  527 IF(AMMP)529,529,529
  528 DELER=AIX(K+1)+(U(K+1)**2+V(K+1)**2)/2.0
      GO TO 537
  529 IF(AMMY)530,531,531
  530 DELER=DELE3
      GO TO 536
  531 IF(GAMC(J))532,533,533
  532 DELER=SIGC(J)
      GO TO 536
  533 IF(AMPY)535,535,534
  534 DELER=DELET
      GO TO 536
  535 DELER=AIX(K)+(U(K)**2+V(K)**2)/2.0
  536 TIC=1.0
  537 DELEK=DELEK-AMMP*DELER
  538 IF(TIC)9907,539,550
  550 WS=DELER
      GO TO 999
  539 WS=AIX(K)+(U(K)**2+V(K)**2)/2.0
  999 IF(DELM)998,543,540
  998 IF(AMX(K)*1.E-6+DELM)9906,997,997
  997 DELM=0.0
      GO TO 543
  540 ENK=AMX(K)*WS+DELEK
  541 U(K)=(SIGMU+AMX(K)*U(K))/DELM
  601 V(K)=(SIGMV+AMX(K)*V(K))/DELM
  603 WS=U(K)**2+V(K)**2
  542 AIX(K)=ENK/DELM-WS/2.0
  543 AMX(K)=DELM
      IF(AMX(K))3900,2007,544
 2007 AIX(K)=0.0
      U(K)=0.0
      V(K)=0.0
```

```
  544 GAMC(J)=AMMP
      FLEFT(J)=AMUR
      YAMC(J)=AMVR
      SIGC(J)=DELER
  545 AMMY=AMPY
      AMMU=AMUT
      AMMV=AMVT
      DELEB=DELET
C     DELET=SPECIFIC TOTAL ENERGY OUT OF THE TOP
C     OF CELL IN QUESTION
C     AMPY=MASS(IN GRAMS)OUT OF THE TOP
C     OF CELL IN QUESTION
  546 K=K+IMAX
  547 CONTINUE
 6802 GO TO 548
 9901 NK=300
      GO TO 9999
 9900 NK=302
      GO TO 9999
 9903 NK=305
      GO TO 9999
 9904 NK=506
      GO TO 9999
 9905 NK=500
      GO TO 9999
 9906 NK=513
      GO TO 9999
 9911 NK=8833
      GO TO 9999
 9908 NK= 17
      GO TO 9999
 9909 NK= 22
      GO TO 9999
 9910 NK= 47
      GO TO 9999
 9907 NK=538
 9999 NR=4
      CALL ERRER(NR,NK)
  548 SUM=0.0
 2005 DO 2001 I=1,IMAX
      K=I+1
      DO 2000 J=1,JMAX
 2008 IF(AIX(K))2004,2000,2000
 2004 SUM=SUM+AIX(K)*AMX(K)
      AIX(K)=0.0
 2000 K=K+IMAX
 2001 CONTINUE
 2003 ETH=ETH-SUM
      Z(104)=Z(104)+SUM
 8002 IF(REZ)8001,8001,8003
 8003 GO TO 8001
 8001 RETURN
      END
```

```
      SUBROJTINE REZONE
      RETURN
      END


      SUBROJTINE FIT 1
      WS = 1.2E-3
      WSA=1.E9
      WSB=0.0
      WSC=0.0
      RETURN
      END



      SUBROJTINE STEADY
C
C                D   I   M   E   N   S   I   O   N
C
      DIMENSION AIX(3500),AMX(3500),P(3500),
     1 U(3500),V(3500),X(60),Y(100),Z(150),
     2 IZ(150),TAU(60),PL(100),UL(100),PR(100),
     3 UR(100),IW1(15),FLEFT(100),YAMC(100),
     4 SIGC(100),GAMC(100)
C
C                C       O       M       M       O       N
C
      COMMON        Z       ,XO       ,X       ,Y
      COMMON        AIX     ,AM       ,AMX     ,AREA     ,E
      COMMON        FS      ,I        ,II      ,IR       ,IWS
      COMMON        J       ,K        ,KP      ,L        ,M
      COMMON        MZ      ,N        ,NK      ,NO       ,NOP
      COMMON        NR      ,P        ,PABOVE  ,PBLO     ,PIOTS
      COMMON        PRR     ,RC       ,REZ     ,RHO      ,RR
      COMMON        SIG     ,TAU      ,TAUOTS  ,TTX      ,TTY
      COMMON        U       ,URR      ,UT      ,UTEF     ,UU
      COMMON        UVMAX   ,V        ,VABOVE  ,VBLO     ,VEL
      COMMON        VTEF    ,WS       ,WSA     ,WSB      ,WSC
      COMMON        XLF     ,XR       ,YLW
      COMMON        YU      ,UR       ,PR
C
C                E   Q   U   I   V   A   L   E   N   C   E
C
      0EQUIVALENCE      (Z,IZ,PROB),      (Z(2),CYCLE),     (Z(3),DT),
     1(Z(4),PRINTS),    (Z(5),PRINTL),    (Z(6),DUMPT7),    (Z(7),CSTOP),
     2(Z(8),PIDY),      (Z(9),TMZ),       (Z(10),GAM),      (Z(11),GAMD),
     3(Z(12),GAMX),     (Z(13),ETH),      (Z(14),FFA),      (Z(15),FFB),
     4(Z(16),TMDZ),     (Z(17),TMXZ),     (Z(18),XMAX),     (Z(19),TXMAX),
     5(Z(20),TYMAX),    (Z(21),AMDM),     (Z(22),AMXM),     (Z(23),DNN),
     6(Z(24),DMIN),     (Z(25),FEF),      (Z(26),DTNA),     (Z(27),CVIS),
     7(Z(28),NPR),      (Z(29),NPRI),     (Z(30),NC),       (Z(31),NPC),
     8(Z(32),NRC),      (Z(33),IMAX),     (Z(34),IMAXA),    (Z(35),JMAX),
     9(Z(36),JMAXA),    (Z(37),KMAX),     (Z(38),KMAXA),    (Z(39),NMAX)
      0EQUIVALENCE      (Z(40),ND),       (Z(41),KDT),      (Z(42),IXMAX),
     1(Z(43),NOD),      (Z(44),NOPR),     (Z(45),NIMAX),    (Z(46),NJMAX),
```

```
     2(Z(47),I1),        (Z(48),I2),        (Z(49),I3),        (Z(50),I4),
     3(Z(51),N1),        (Z(52),N2),        (Z(53),N3),        (Z(54),N4),
     4(Z(55),N5),        (Z(56),N6),        (Z(57),N7),        (Z(58),N8),
     5(Z(59),N9),        (Z(60),N10),       (Z(61),N11),       (Z(62),NRM),
     6(Z(63),IRAD),      (Z(64),XNRG),      (Z(65),SN),        (Z(66),DXN),
     7(Z(67),RADER),     (Z(68),RADET),     (Z(69),RADEB),     (Z(70),DTRAD),
     8(Z(71),REZFCT),    (Z(72),RSTOP),     (Z(73),SHELL),     (Z(74),BBOUND),
     9(Z(75),TOZONE),    (Z(76),ECK),       (Z(77),SBOUND),    (Z(78),X1)
     0EQUIVALENCE        (Z(79),X2),        (Z(80),Y1),        (Z(81),Y2),
     1(Z(82),CABLN),     (Z(83),VISC),      (Z(84),T),         (Z(85),GMAX),
     2(Z(86),WSGO),      (Z(87),WSGX),      (Z(88),GMADR),     (Z(89),GMAXR),
     3(Z(90),S1),        (Z(91),S2),        (Z(92),S3),        (Z(93),S4),
     4(Z(94),S5),        (Z(95),S6),        (Z(96),S7),        (Z(97),S8),
     5(Z(98),S9),        (Z(99),S10),
     6(Z(127),DX),       (Z(128),DY1),      (Z(129),DY),
     7(Z(130),NFITB),    (Z(131),NFITT),    (Z(132),NFITR),
     8(Z(133),NPARB),    (Z(134),NPART),    (Z(135),NPARR)
C
      EQUIVALENCE        (P,IW1)
      EQUIVALENCE(P(200),GAMC),(P(400),YAMC)
      EQUIVALENCE        (UR,UL,FLEFT),     (PR,PL,SIGC)
C
      VEL = 0.0
      FLAG2 = 12.0
   20 FLAG1 = 0.0
   30 READ (4) PR(1), PR(2), N3
      WRITE (6,8006) PR(1),PR(2),N3
      IF (PR(1).EQ.555.0) 40,31
   31 IF (PR(1).EQ.666.0) 32,35
   32 WRITE (6,8008)
      GO TO 1000
   35 FLAG1 = FLAG1 + 1.0
      IF (FLAG1 - FLAG2) 34,34,9901
   34 WRITE (6,8100) FLAG1
      GO TO 30
   40 READ (4) (Z(I),I=1,3)
      WRITE (6,8104) (Z(I),I=1,3)
      READ (4) (DUM,DUM,DUM,DUM,P(I),I = 1,KMAXA)
      WRITE (6,8106) (P(I),I=1,3)
      READ (4) DUM
      READ (4) DUM
      IF (VEL) 50,50,60
   50 DO 55 I = 1,KMAX
   55 AMX(I) = P(I)
      VEL = 1.0
      WRITE (6,8102)
      GO TO 30
   60 SUM = 0.0
      I = 3
   70 DO 75 J = 2,JMAX
      K = I + 1 + (J - 1)*IMAX
      DIFF = ABS(P(K)) - ABS(AMX(K))
   75 SUM = SUM + ABS(DIFF)
   80 WRITE (6,8002) PR(2), SUM
      IF (CSTOP - PR(2)) 1000,1000,90
   90 DO 95 I = 1,KMAX
   95 AMX(I) = P(I)
```

```
      GO TO 20
9901  WRITE (6,8001) FLAG2
      GO TO 1000
8001  FORMAT (22H0555 WAS NOT FOUND IN ,F5.1,7H CYCLES )
8002  FORMAT (1740CYCLE NUMBER IS ,F5.0,5X,22HSUM OF DIFFERENCES IS ,
     1  0PE12.6 //)
8008  FORMAT (16H0666.0 WAS FOUND  /)
8006  FORMAT(10X,16HFIRST RECORD IS ,2E12.4,I6,/)
8100  FORMAT (10X,9HFLAG1 IS ,F5.1,/)
8102  FORMAT (10X,14HVEL SET TO 1.0  /)
8104  FORMAT (10X,25HTHE FIRST 3 Z VALUES ARE ,3E12.6,/)
8106  FORMAT (10X,25HTHE FIRST 3 P VALUES ARE ,3E12.6,/)
1000  REWIND 4
      RETURN
      END



      SUBROUTINE ERRER(NR,NK)
      CALL TIME
      WRITE (6,8880) NR,NK
8880  FORMAT (10X,5H NR= ,I2,5X,5H NK= ,I4,/)
      CALL EXIT
      END
```

APPENDIX B

ADDITIONAL PLOTS FOR THE IDEALIZED PROBLEM

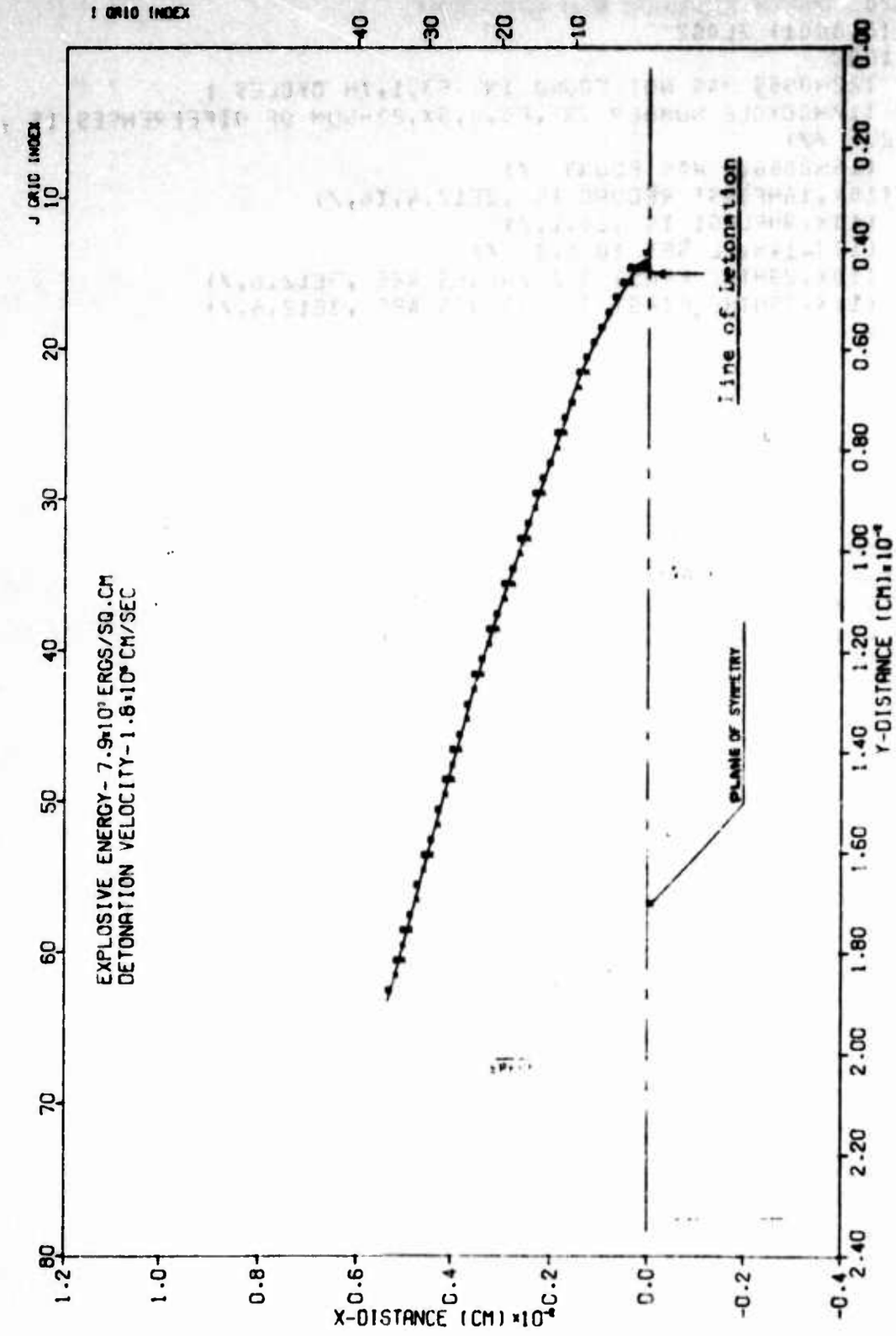


Figure B1. Shock Front Location in the XY Plane

SHELLTC CALCULATION OF POINT SOURCE IN HYPERSONIC FLOW
TIME 0.001672 SEC CYCLE 310 PROBLEM 22.000

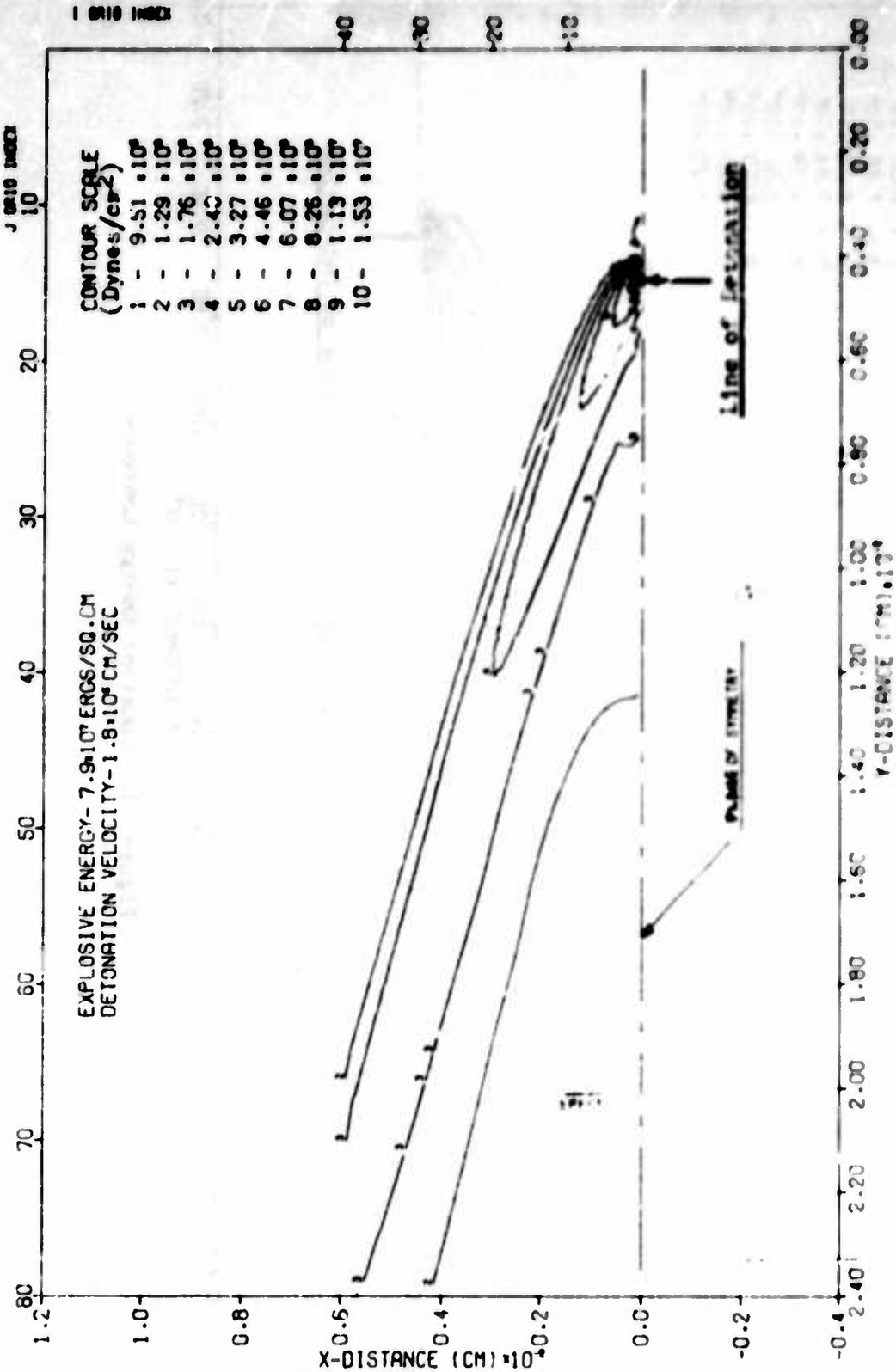


Figure B2. Constant Pressure Contours (Isobars)

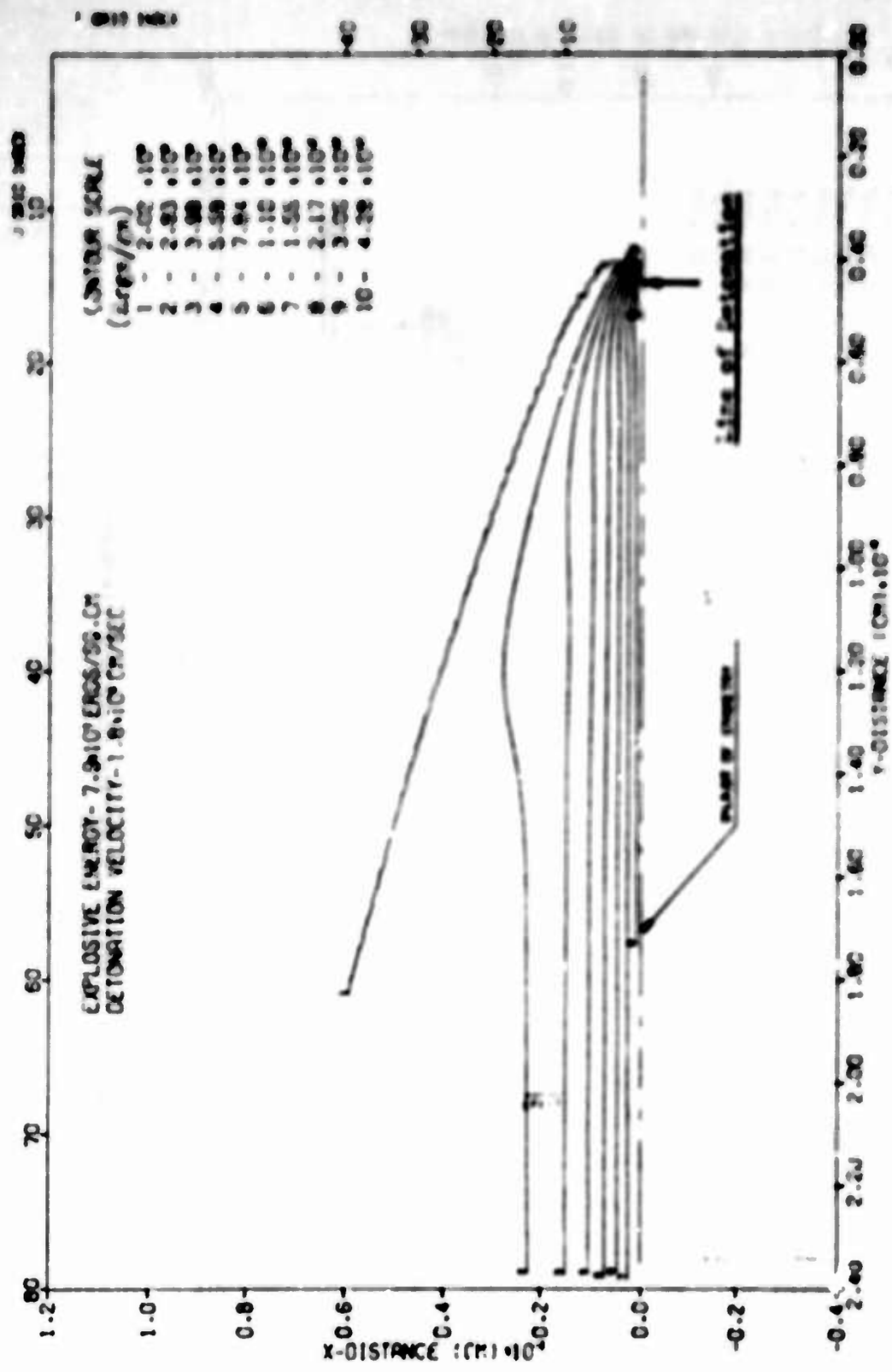


Figure 33. Constant Energy Contours

SHELLTC CALCULATION OF POINT SOURCE IN HYPERSONIC FLOW
TIME 0.001672 SEC CYCLE 310 PROBLEM 22.000

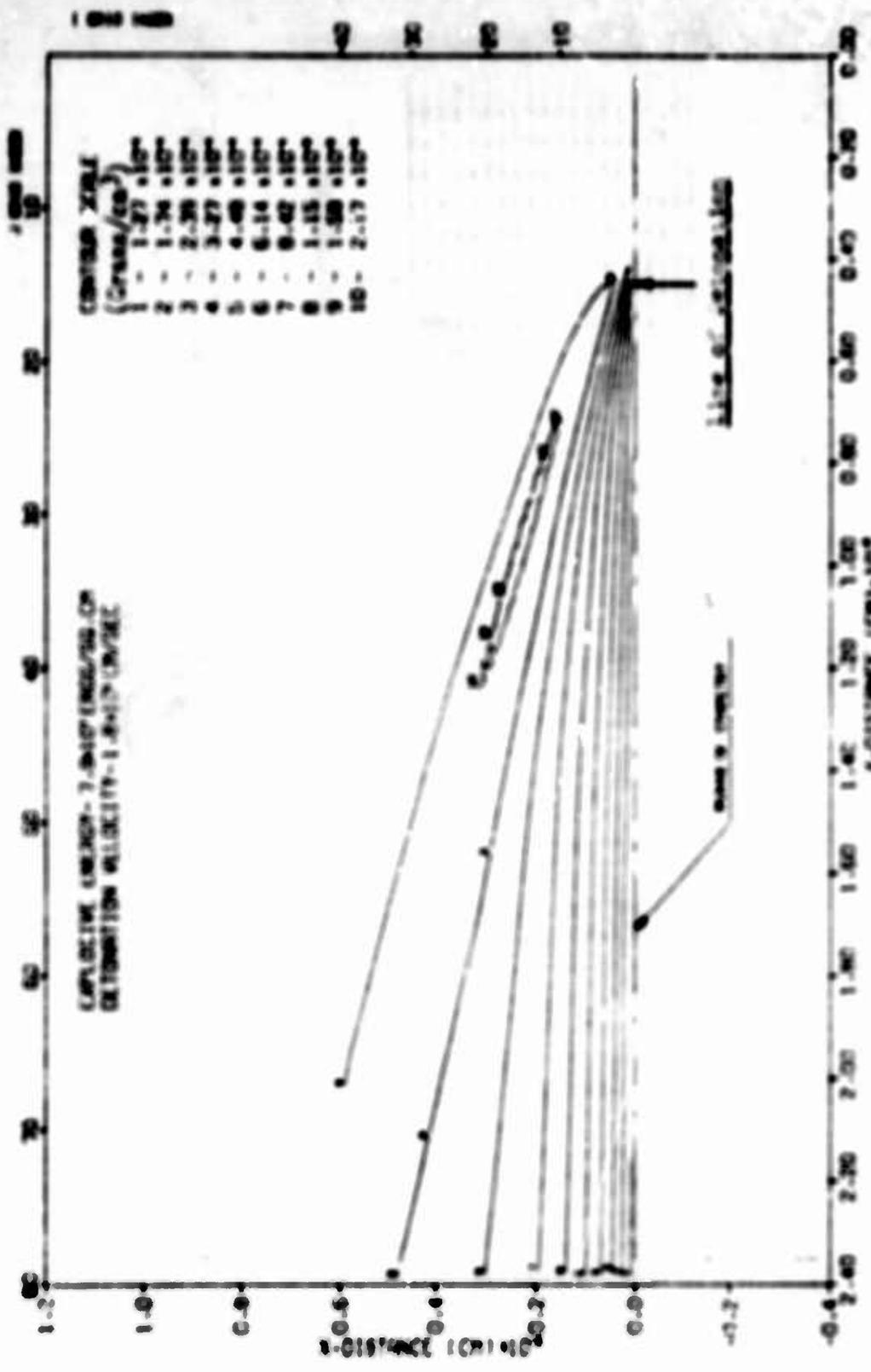


Figure 24. Constant Density Contours

SHELLIC CALCULATION OF POINT SOURCE IN HYPERSONIC FLOW
TIME 0.001672 SEC CYCLE 310 PROBLEM 22.000

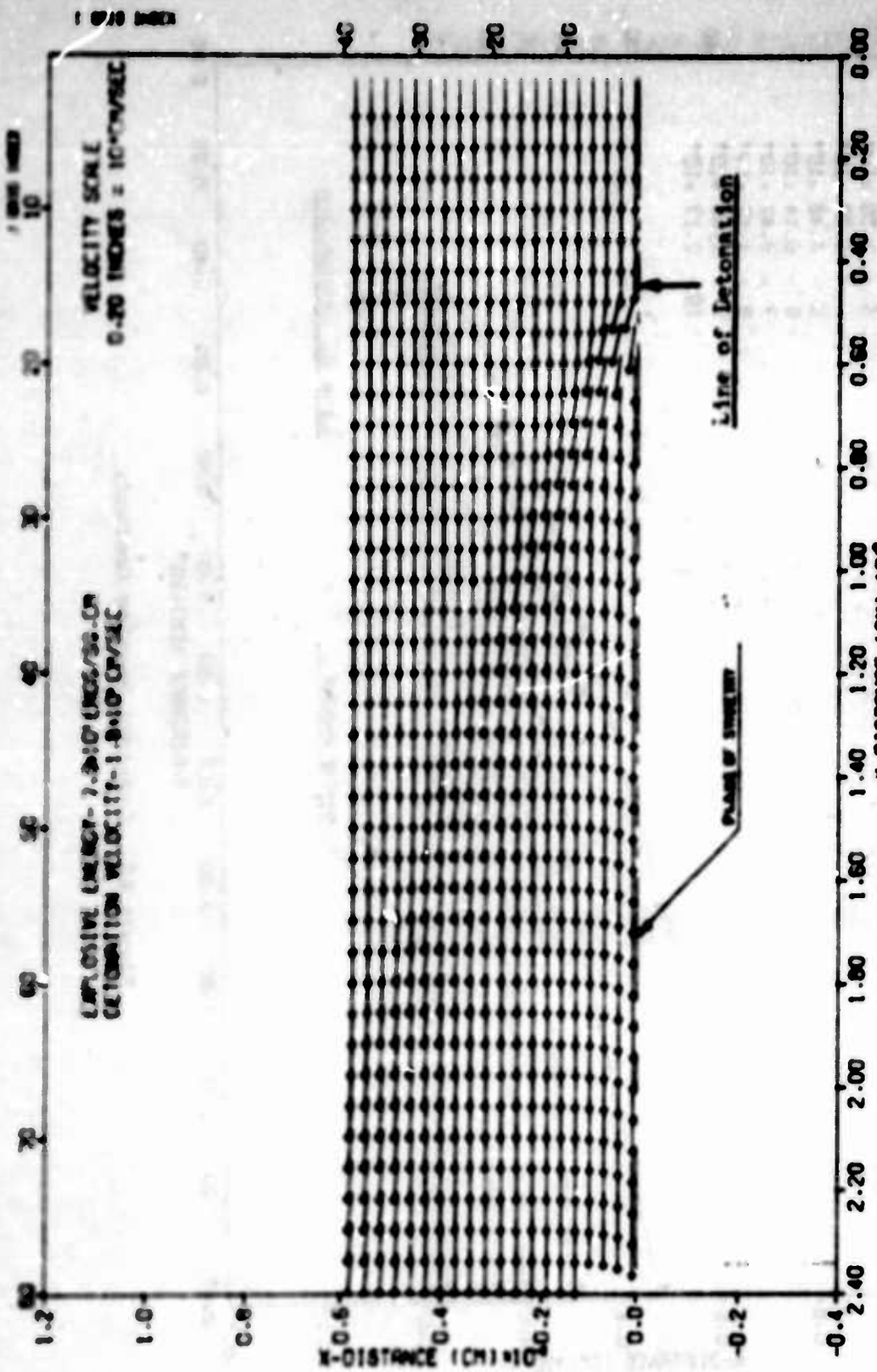


Figure E5. Velocity Vector Plot

SHELLTC CALCULATION OF POINT SOURCE IN HYPERSONIC FLOW
TIME 0.001672 SEC CYCLE 310 PROBLEM 22.000

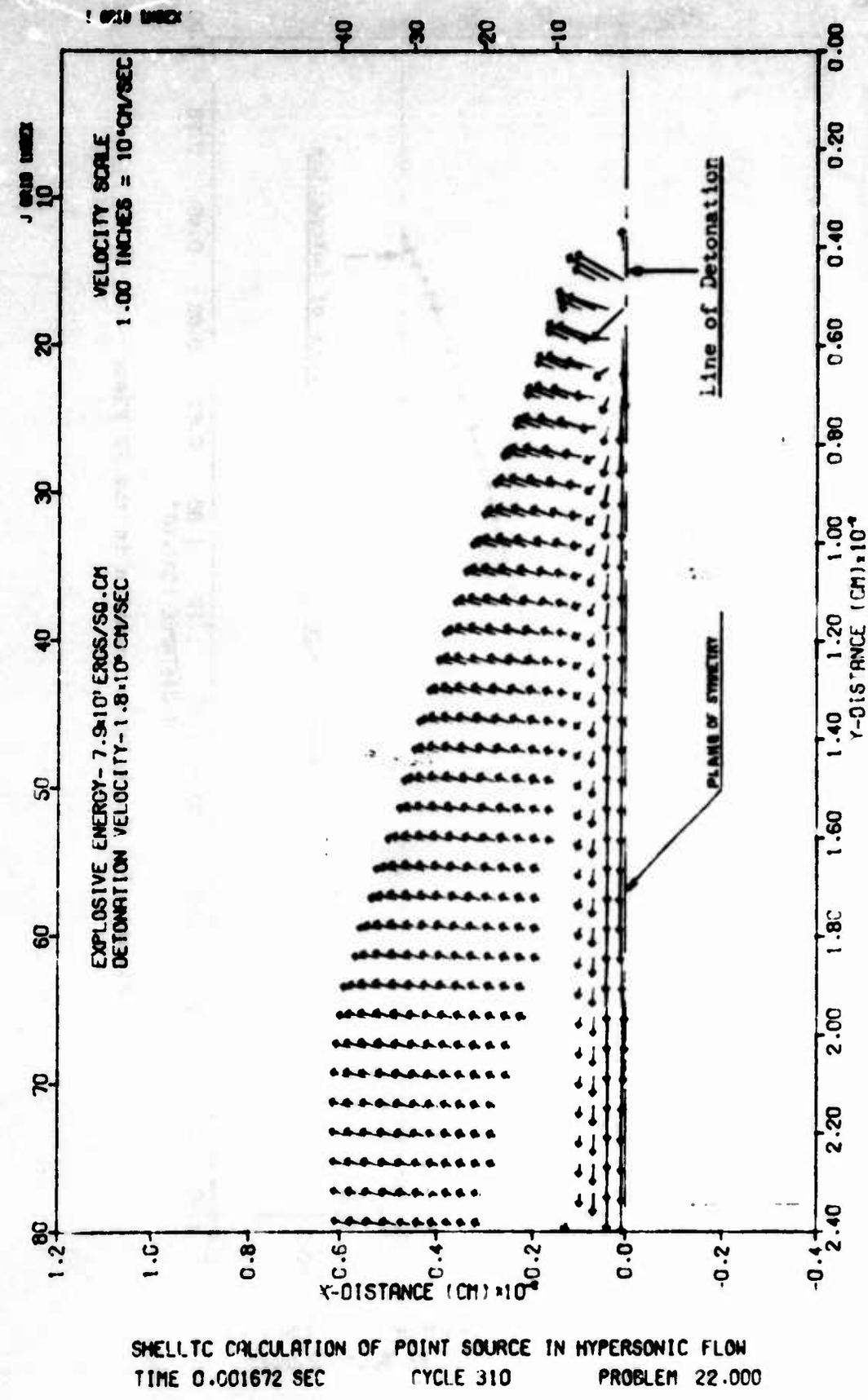


Figure B6. Plot of Velocity Vector with Background Velocity Subtracted

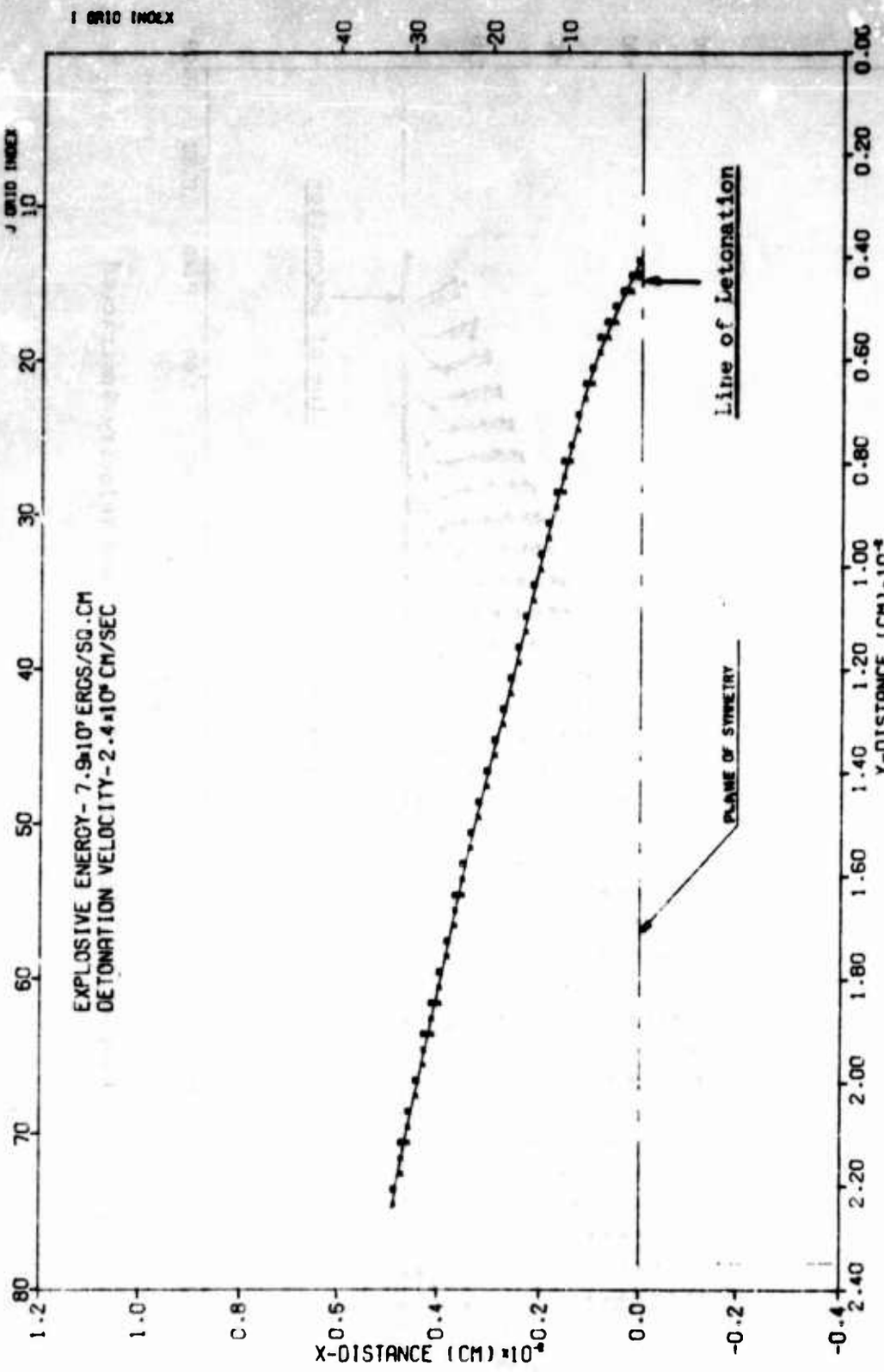


Figure B7. Shock Front Location in the XY Plane

SHELLTC CALCULATION OF POINT SOURCE IN HYPERSONIC FLOW
TIME 0.001199 SEC CYCLE 430 PROBLEM 22.000

NOT REPRODUCIBLE

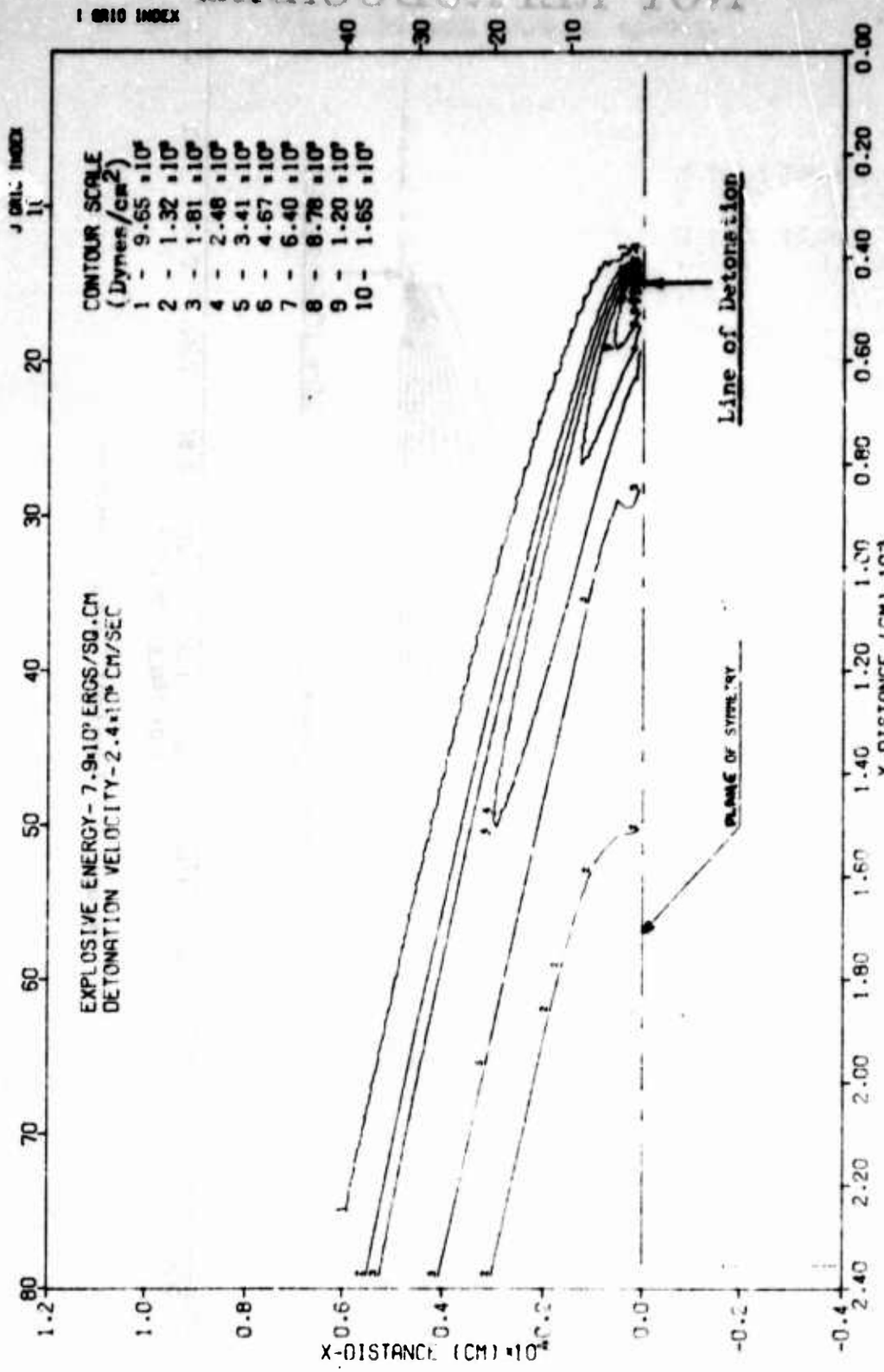


Figure 88. Constant Pressure Contours (Isobars)

NOT REPRODUCIBLE

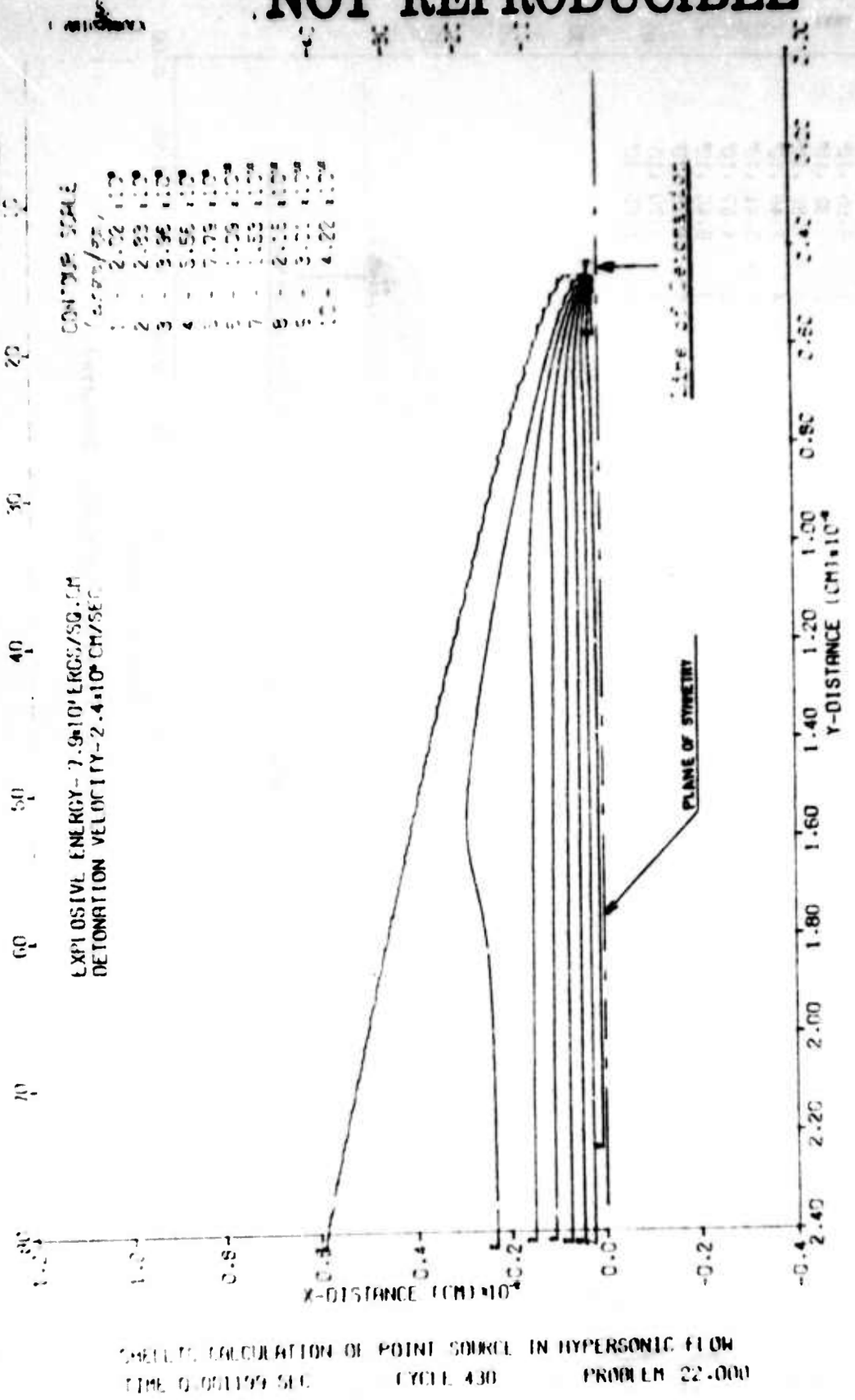


Figure 89. Constant Energy Contours

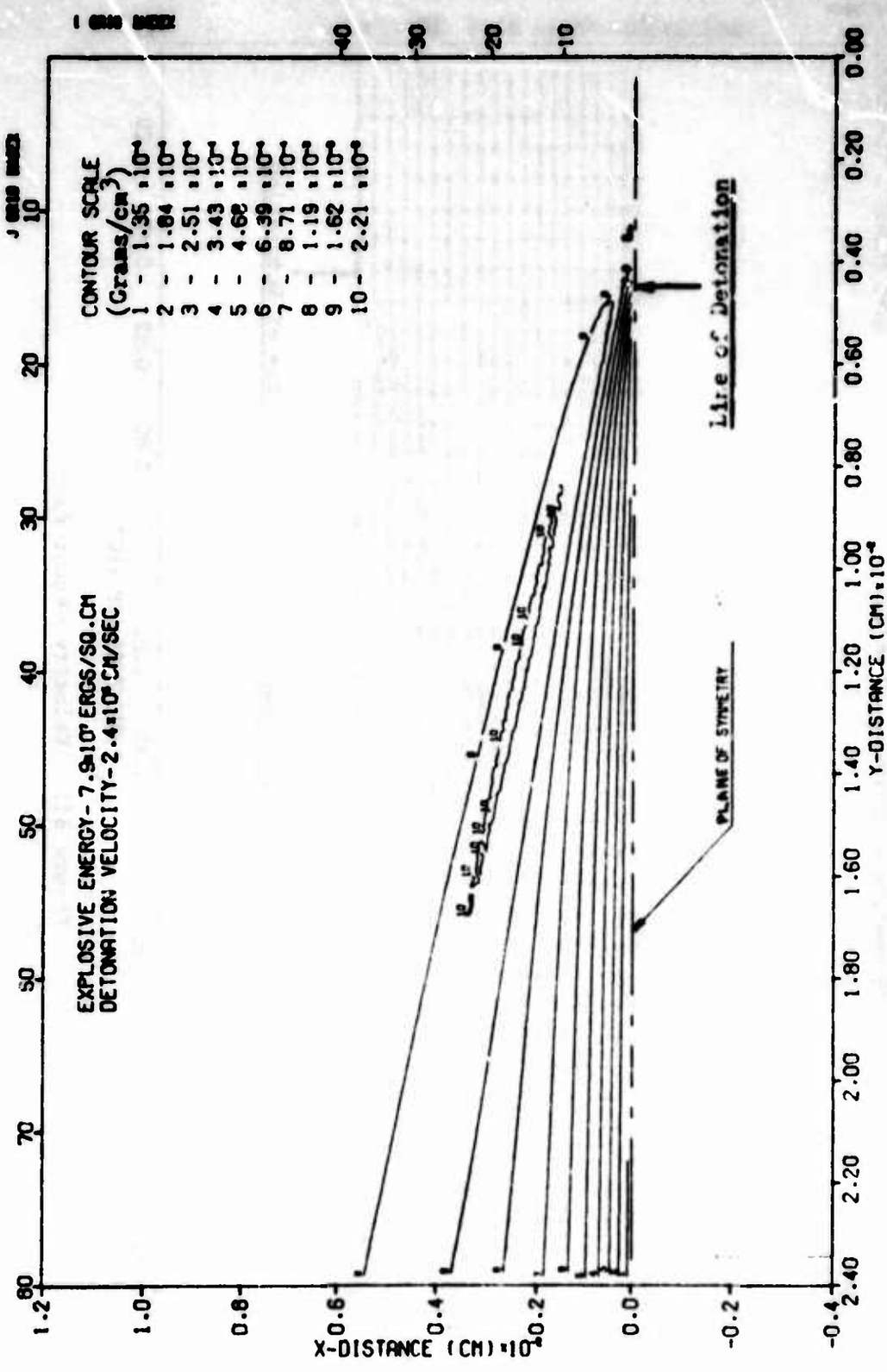


Figure B10. Constant Density Contours

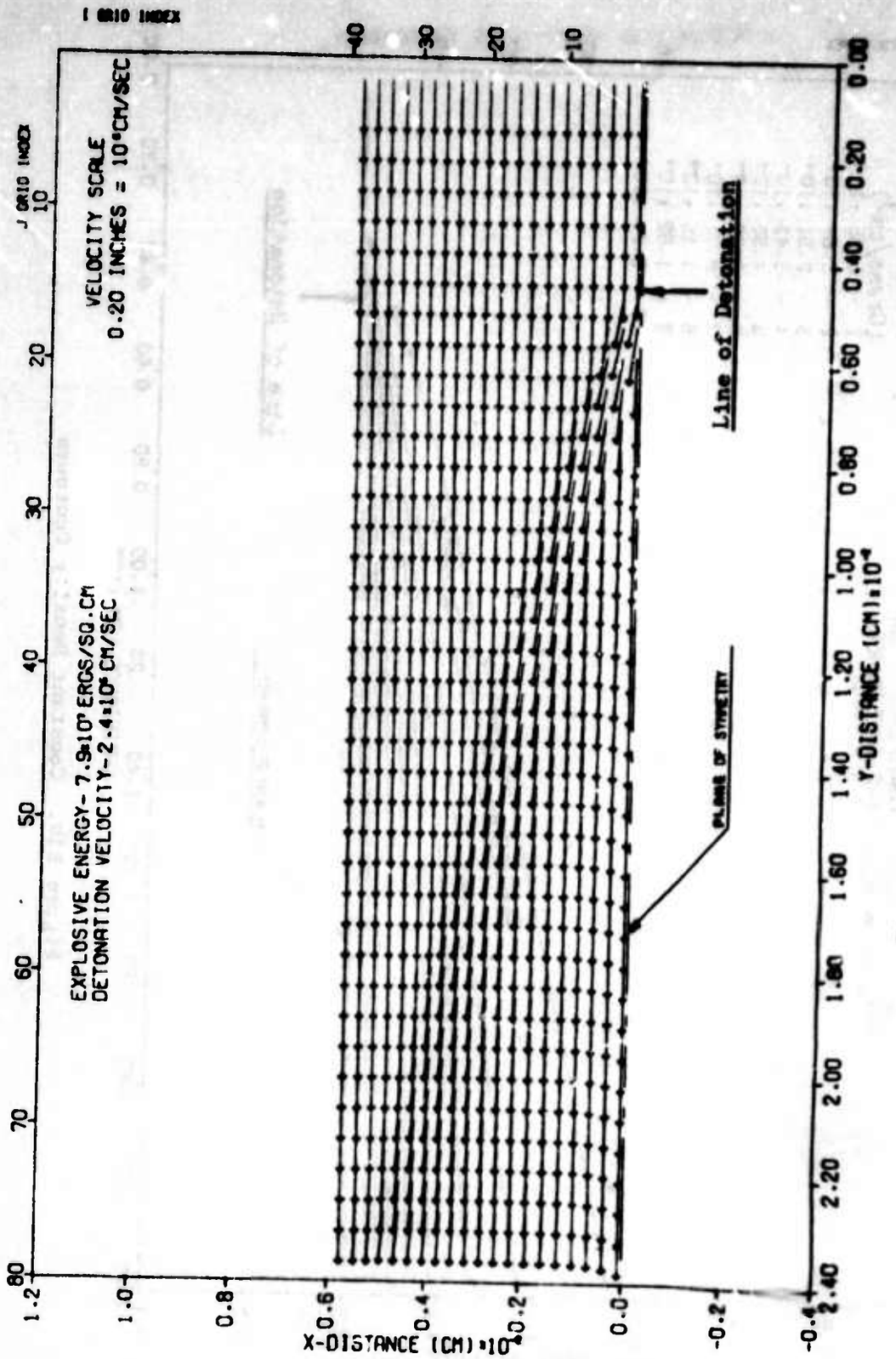


Figure B11. Velocity Vector Plot

SHELLTC CALCULATION OF POINT SOURCE IN HYPERSONIC FLOW
TIME 0.JG1199 SEC CYCLE 430 PROBLEM 22.000

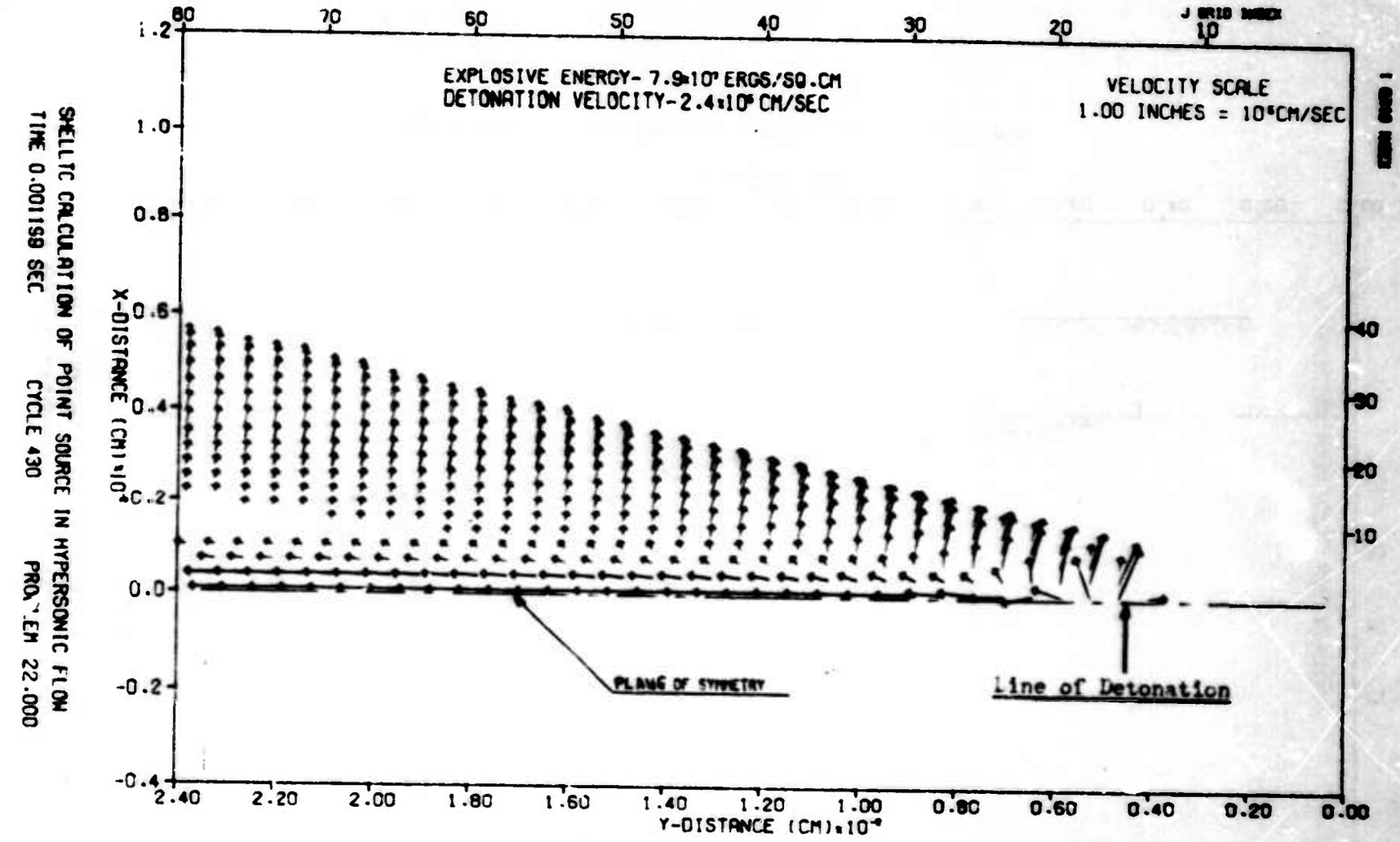


Figure 312. Plot of Velocity Vector with Background Velocity Subtracted

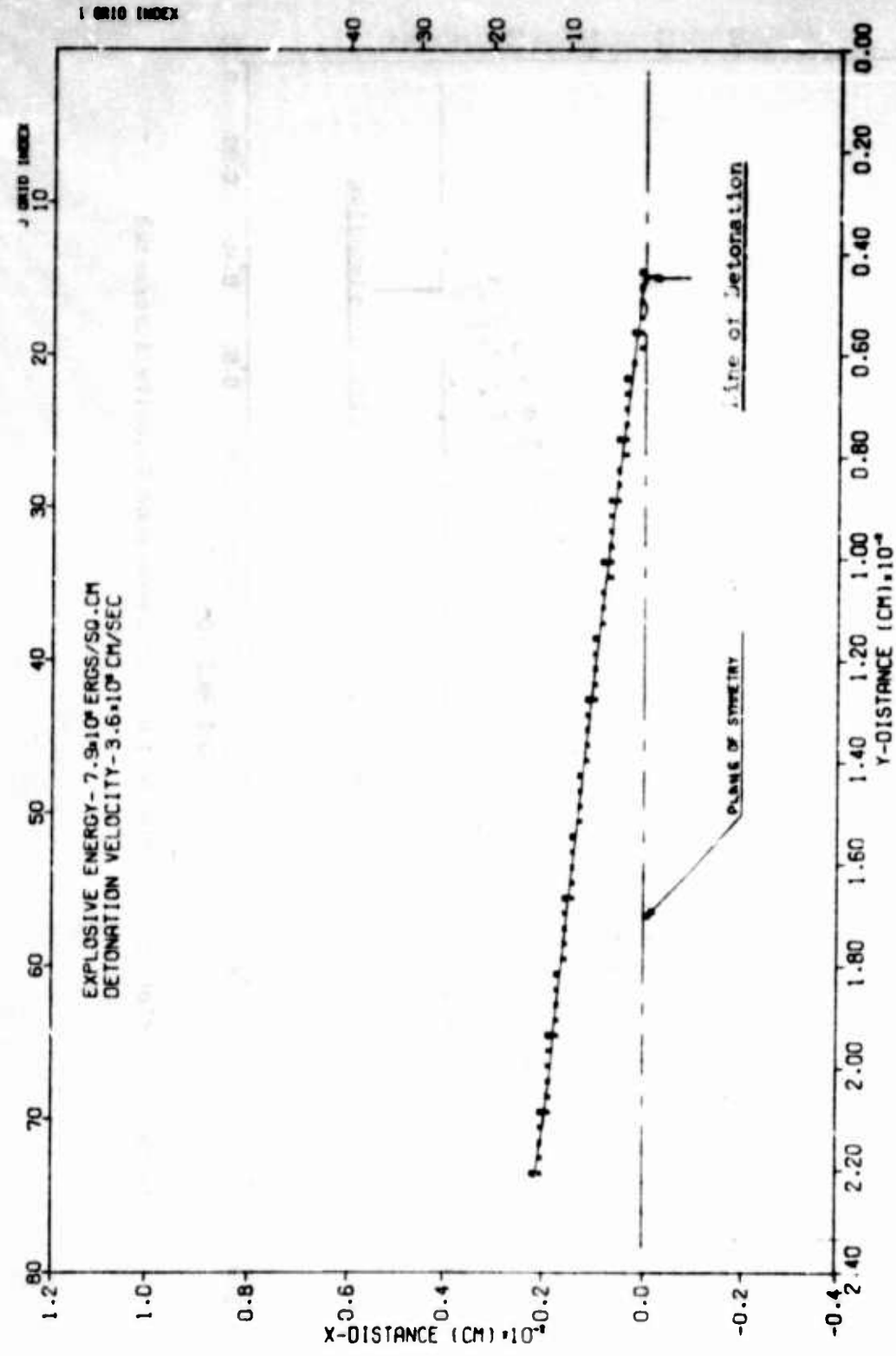


Figure 313. Shock Front Location in the XY Plane

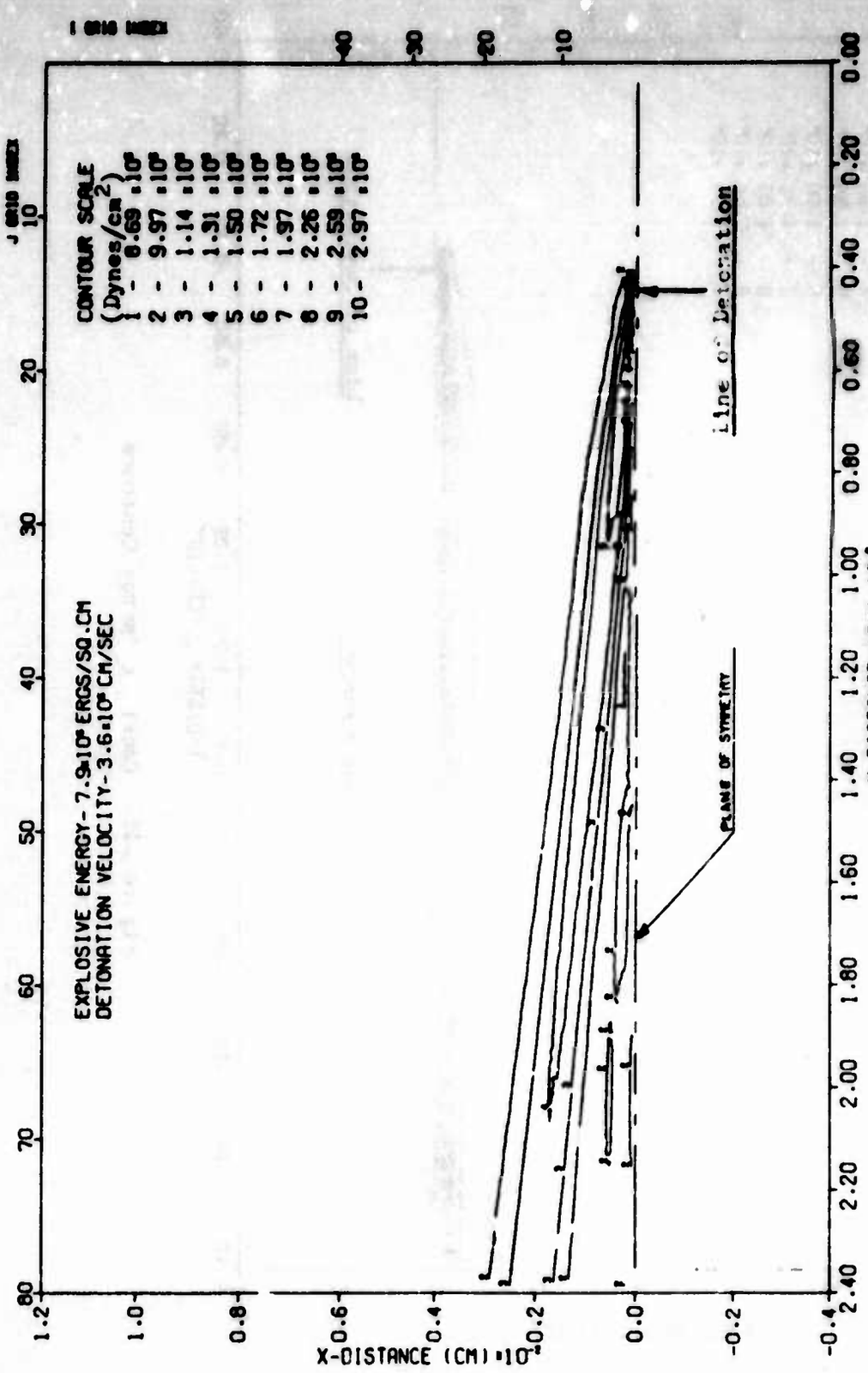


Figure 814. Constant Pressure Contours (Isobars)

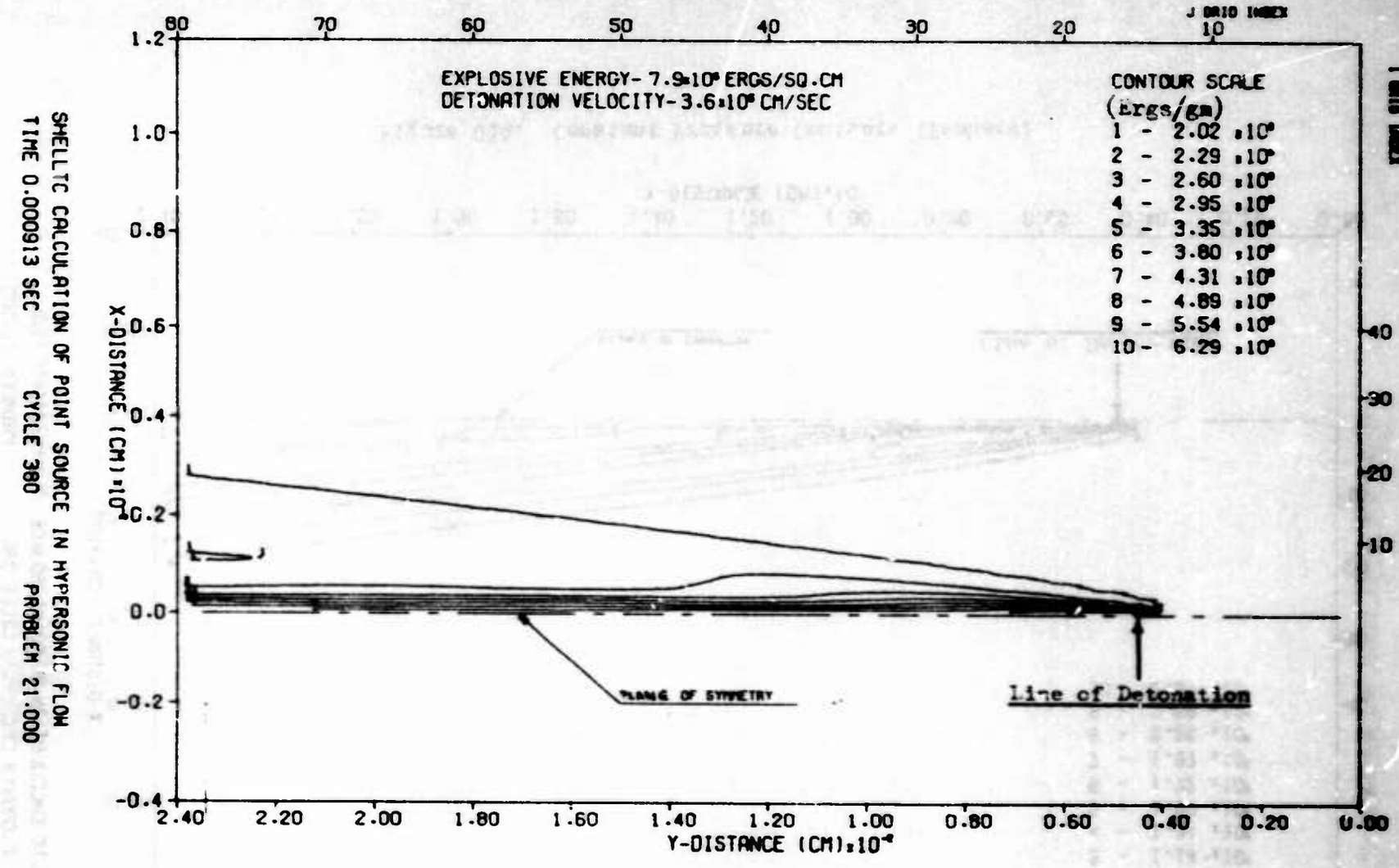


Figure B15. Constant Energy Contours

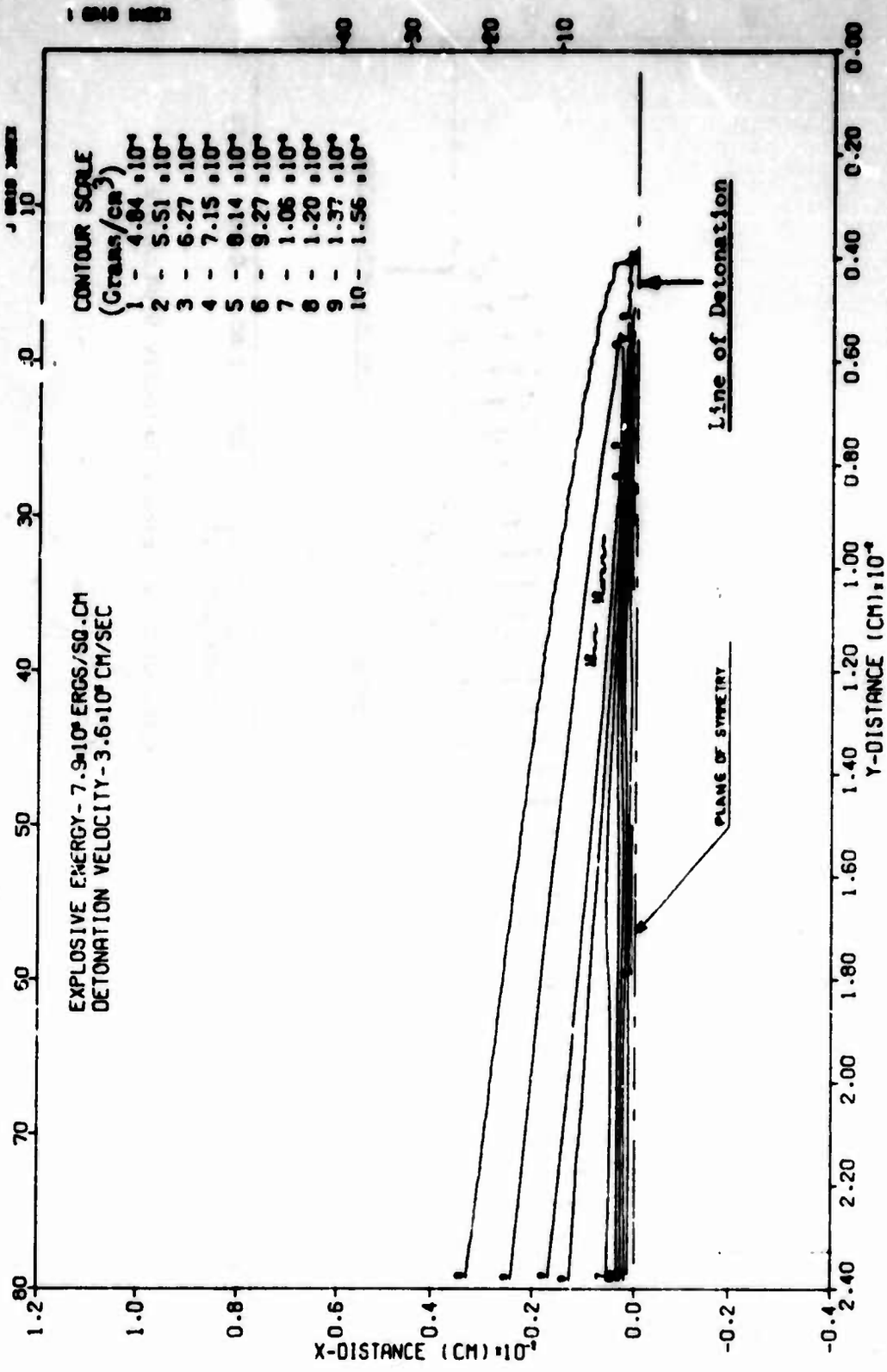


Figure 316. Constant Density Contours

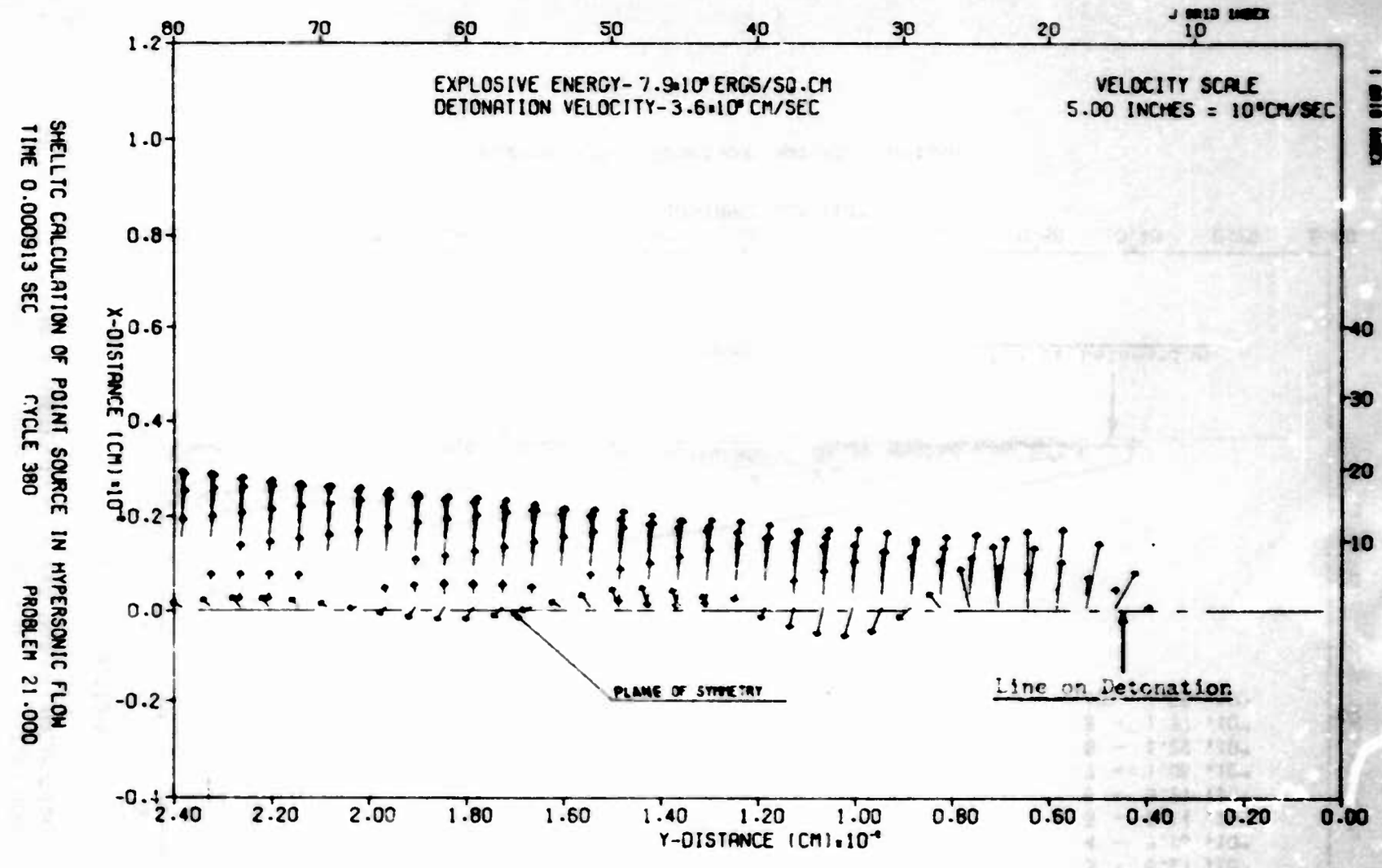


Figure B17. Plot of Velocity Vector with Background Velocity Subtracted

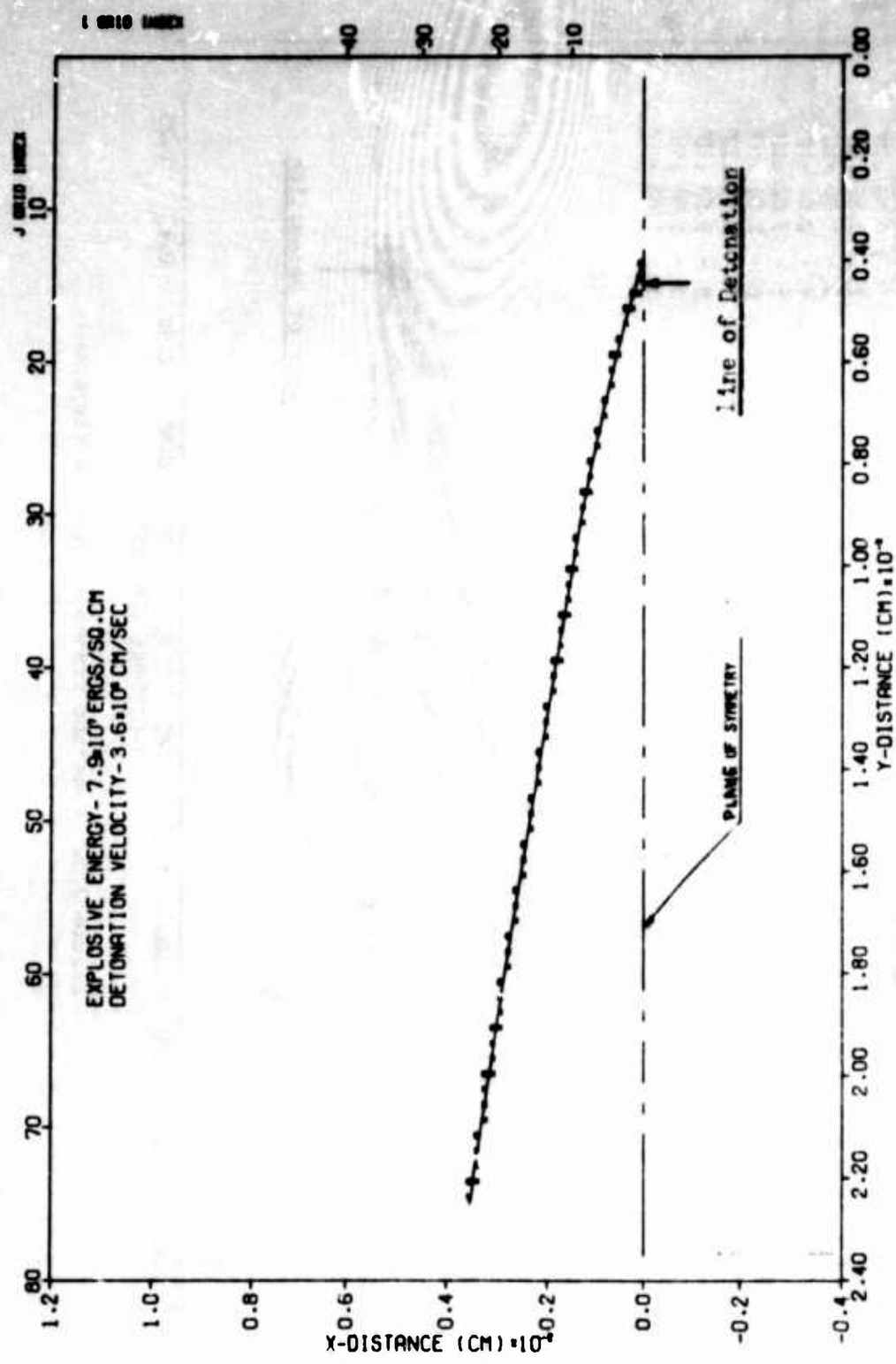


Figure B18. Shock Front Location in the XY Plane

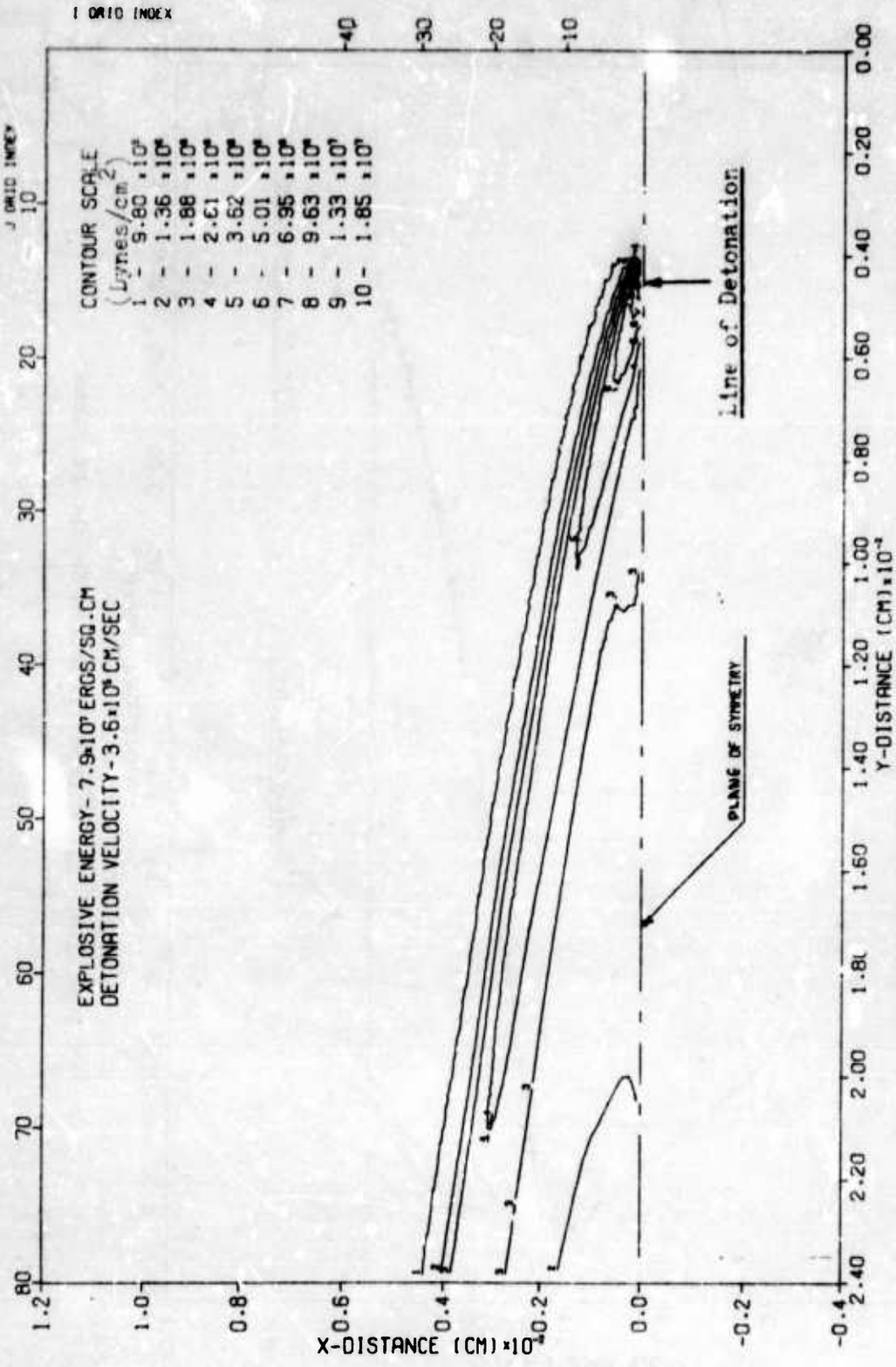


Figure B19. Constant Pressure Contours (Isobars)

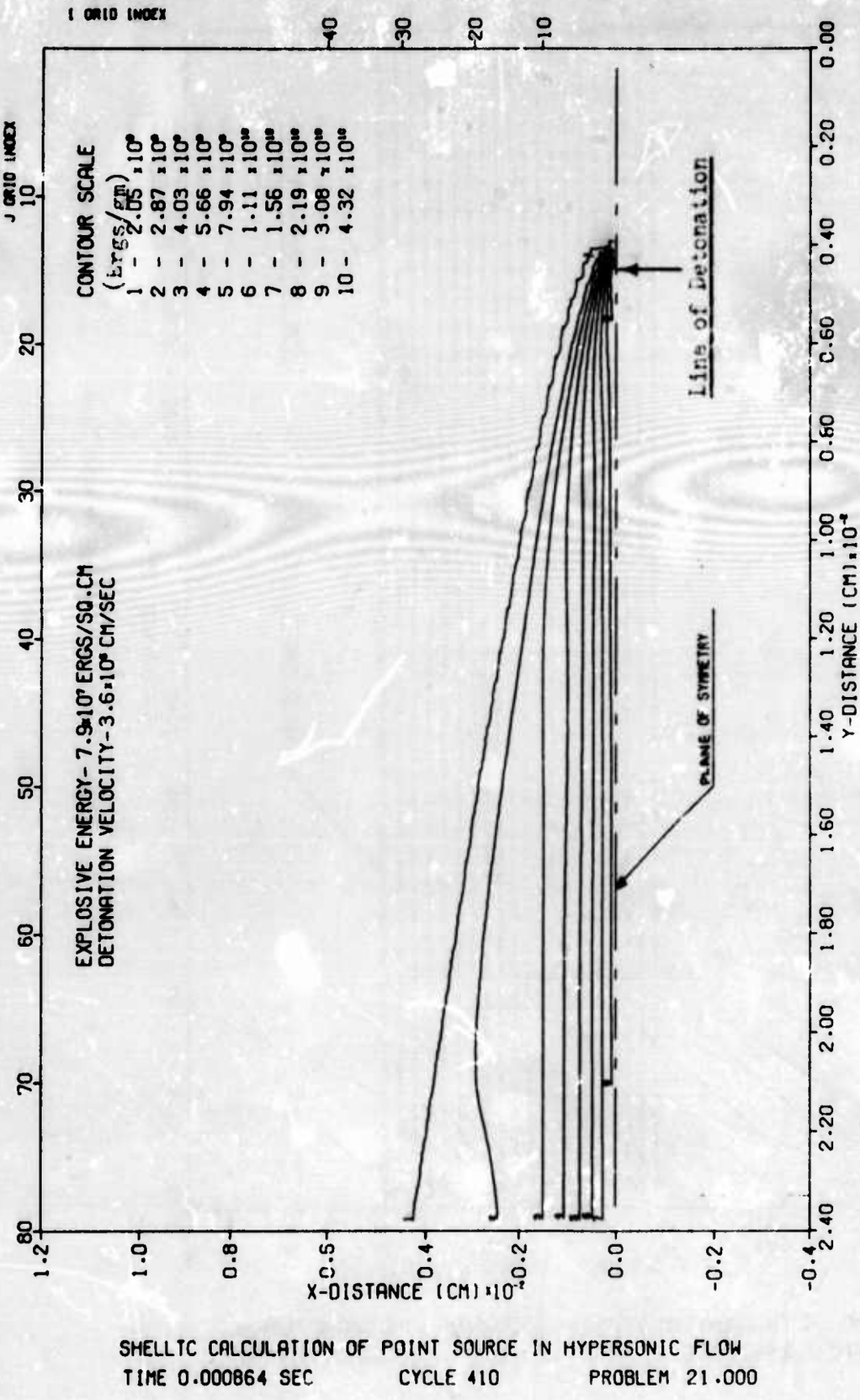


Figure B20. Constant Energy Contours

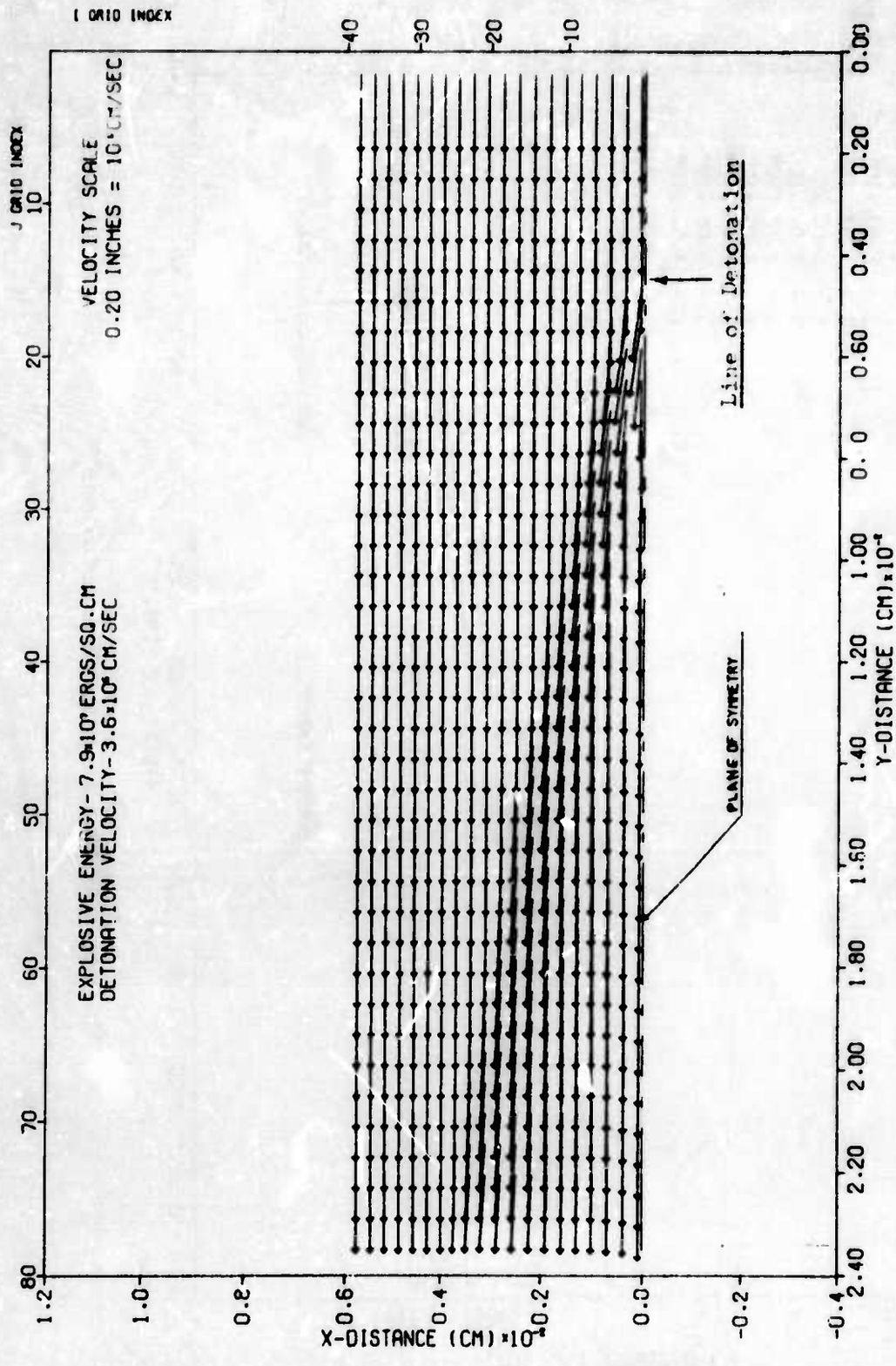


Figure 321. Velocity Vector Plot

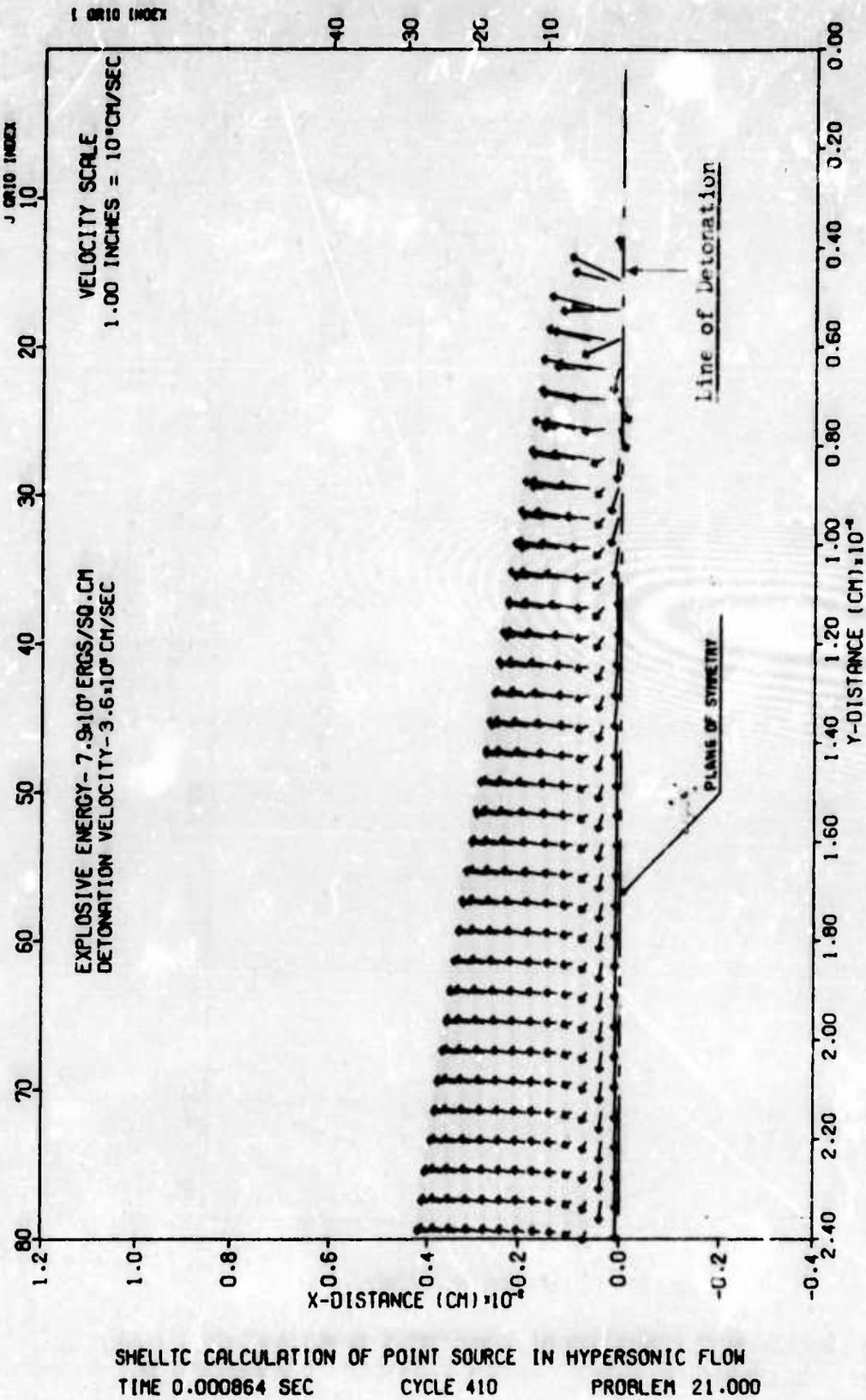


Figure B22. Plot of Velocity Vector with Background Velocity Subtracted

Figure 323. Shock Front Location in the XY Plane

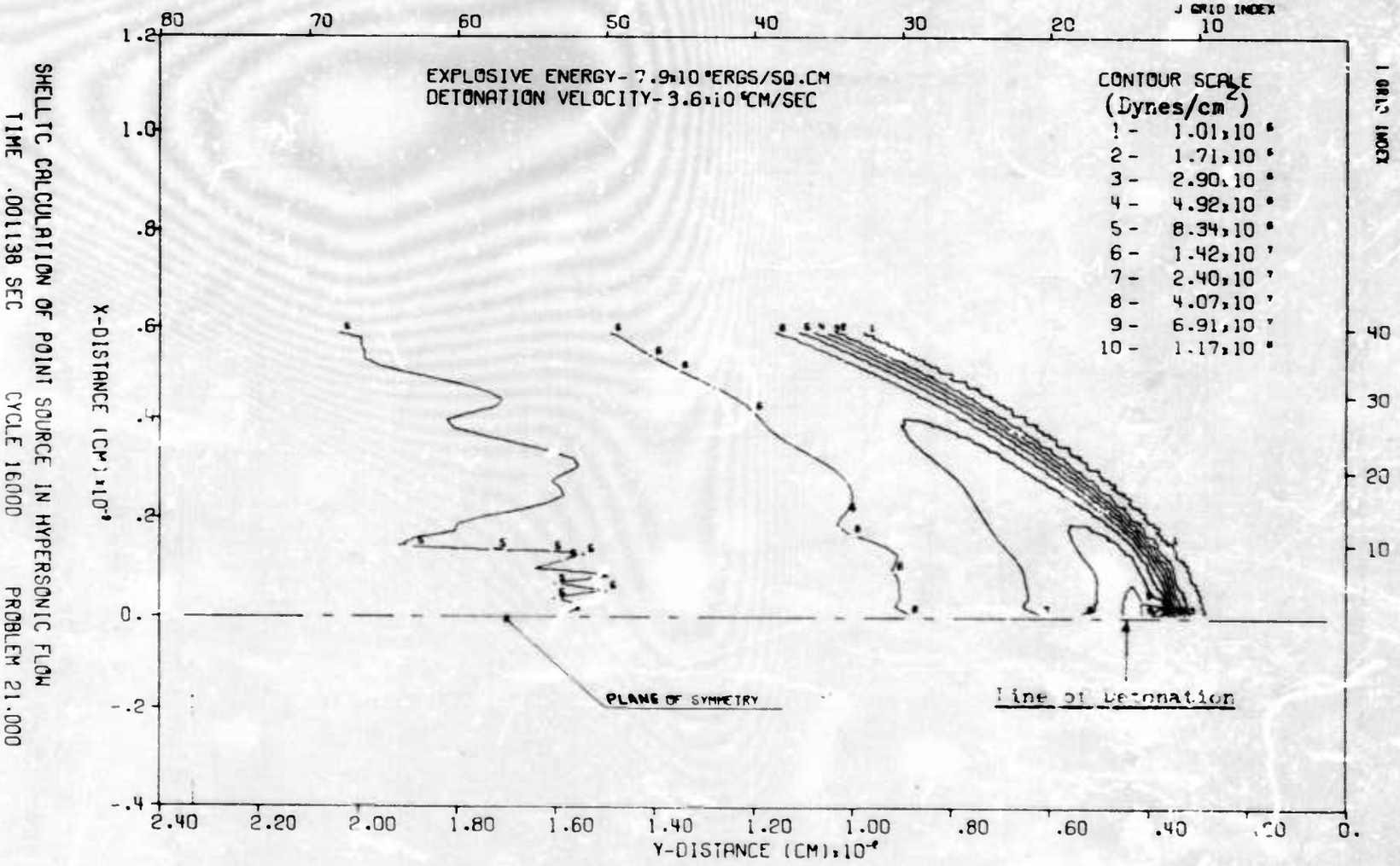


Figure B24. Constant Pressure Contours (Isobars)

Figure 325. Constant Energy Contours

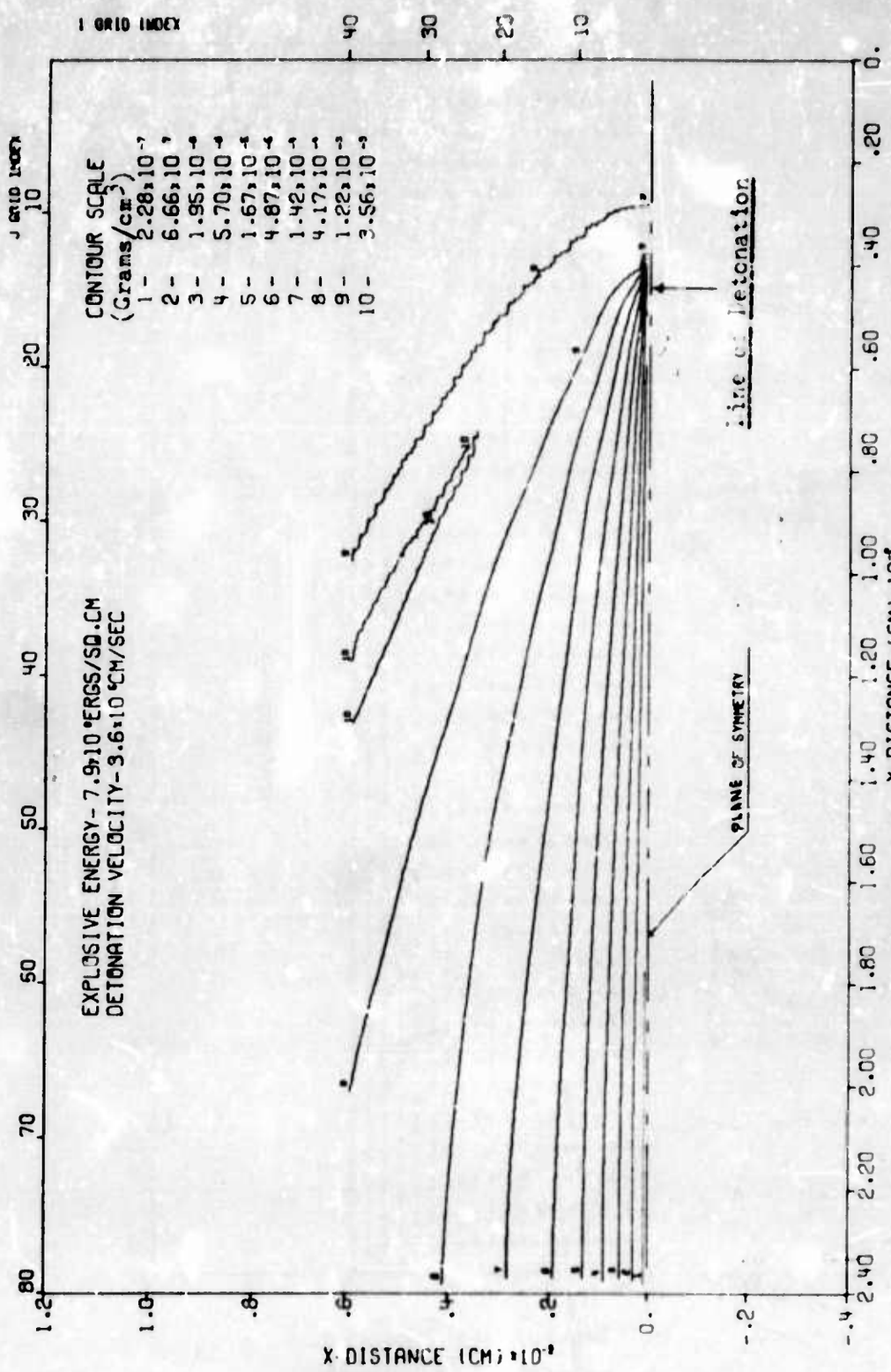


Figure B26. Constant Density Contours

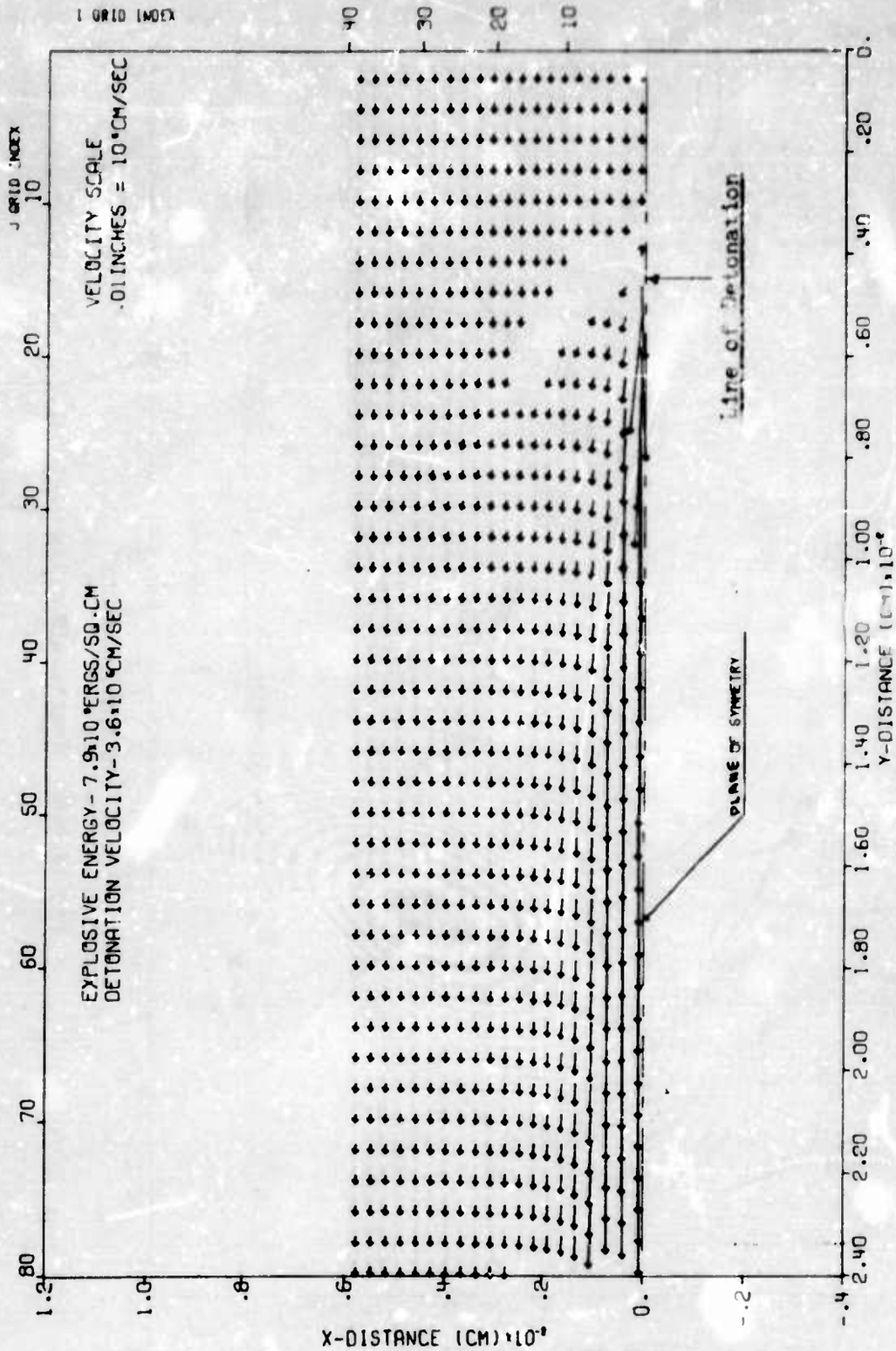


Figure B27. Velocity Vector Plot

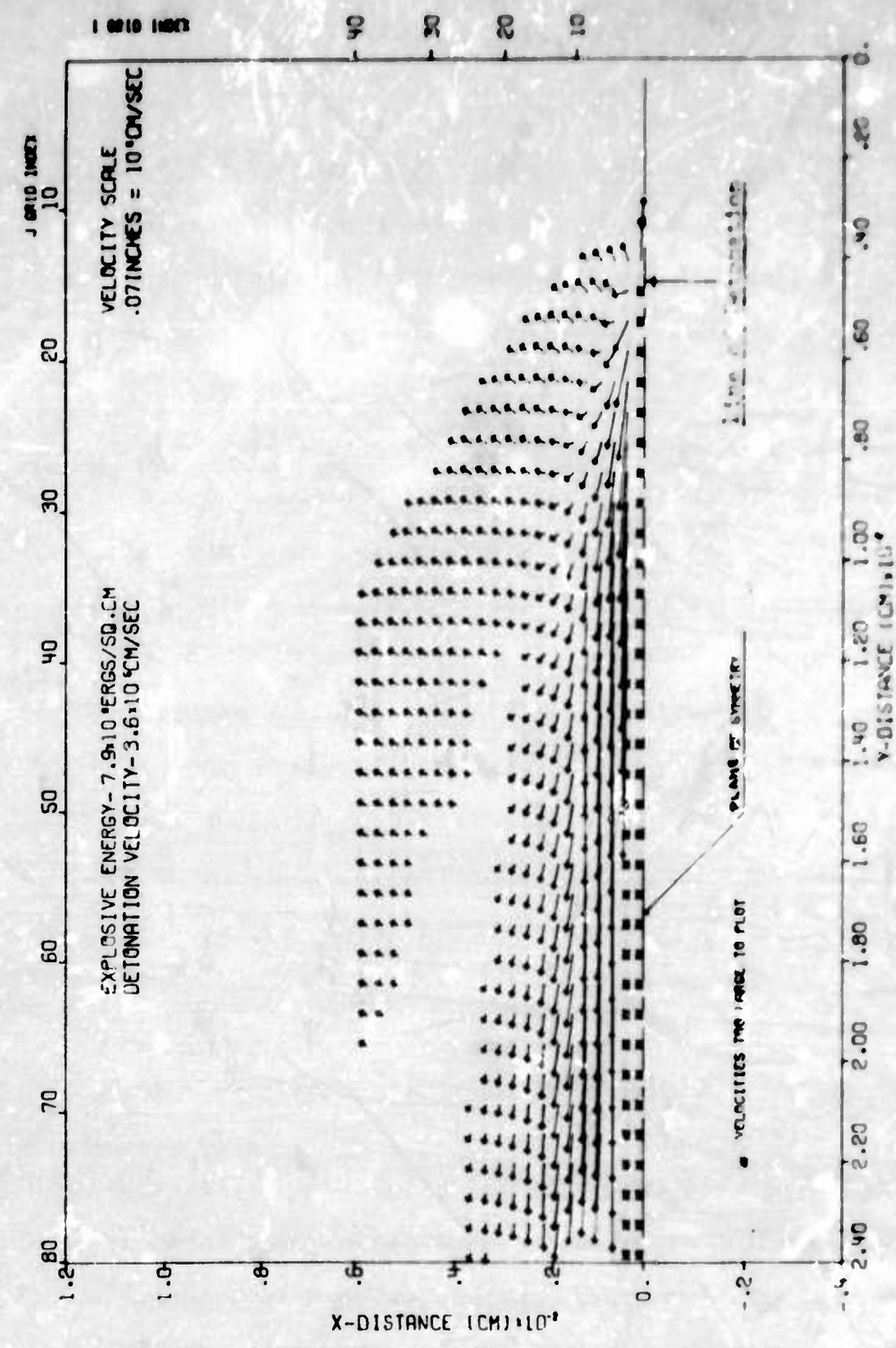


Figure 128. Plot of Velocity Vector with Background Velocity Subtracted

APPENDIX C

THE SHELTC7 CODE

The SHELTC7 code is a seven-material member of the SHELL family of hydrodynamic codes. It was developed for use in determining the flow of seven distinct mass species through an Eulerian grid and can be used with either cylindrical or cartesian coordinates.

SHELTC7 was developed from the RADISH5 code which is a five-material member of the SHELL family. The basic operation of RADISH5 is described in Reference 35. The same basic equations that govern SHELL-OIL (see Appendix A) apply to SHELTC7. In this case, however, the equation of conservation of mass described in Appendix A (Equation A1) is a global continuity equation and refers to the total mass in the problem. In expanding from single material to multi-material, an individual species continuity equation is incorporated for each specie carried by the multi-material code. In a cartesian coordinate system

$$\frac{\partial \rho_i}{\partial t} + \frac{\partial}{\partial x}(\rho_i u) + \frac{\partial}{\partial y}(\rho_i v) = 0 \qquad (i = 1,2,\cdots,7) \qquad (C1)$$

It is assumed here that the total quantity of each individual specie is preserved during the mass transport phase (none is produced or destroyed--no reactions are allowed to occur). The momentum and energy associated with each specie are also conserved during the mass transport phase.

The same basic computational scheme which was used in SHELLTC is employed in SHELTC7. A computational grid is established by an auxiliary generator code (CLAMTC7) and loaded with an initial distribution of seven materials. Initial values of velocity, energy, and temperature are assigned to the total mass in each cell. This information is established as the zero cycle on the problem tape by this generator code. The equation of state subroutine (STATE7) is called from CDT to determine a cell temperature and pressure. The multi-material equation of state is discussed in Section III of the main body of the text. A time increment for the cycle is determined by CDT based on the particle velocity criteria, the same as for SHELLTC (see Appendix A).

The PHASE I computations are also the same in SHELTC7 as in SHELLTC. Only the negative integration option is deleted. In this case, if an unstable situation is encountered (such as negative internal energy), the computation is immediately terminated with an appropriate error message. In Phase I, the mass in each cell is considered to be a single uniform material because multiple materials in the cell do not affect the Phase I calculations.

It is in the Phase II computations that all mass transport occurs and multi-material effects are seen in SHELTC7. First the total mass transport in and out of a particular cell is accomplished much as in the single-material case. Then the individual mass species are transported across the cell boundaries by a weighting scheme which assigns the same

proportions amongst the species in the transported mass as existed in the donor cell just before the mass transport. Conservation of energy is used to determine new cell energies and conservation of momentum is used to compute new velocities. Conservation of momentum and energy is also accomplished for each individual mass specie, and this information is also used in obtaining the new velocities. Once mass transport into and out of any cell is completed, the mass in that cell is considered to be uniformly mixed. This leads to an artificial mixing and diffusion of one mass specie through the others which is discussed in some detail in the main body of the text (Section IV).

The code then accomplishes subroutines INTFACE and PDC (both discussed in Section III, main body of text) before returning to CDT. The INPUT and EDIT subroutines are essentially the same as for SHELLTC except provisions for recording and printing out the quantities of all seven species in each cell are provided.

In SHELTC7 a total of 13 variables must be stored for each cell in the computational grid. Hence, it can only be run on computers with large memory capacities. Further, the time required for execution is substantial (3 to 5 cycles per second) making it imperative to run this code on a fast computer. SHELTC7 is currently structured for running on the CDC 6600 computer. A listing of SHELTC7 as used in this work follows.

LISTING OF SHELTC7 CODE

```
      PROGRAM SHELTC7(TAPE4,INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C
C     ******  NOTE-THIS PROGRAM CHANGED TO ACCOMODATE SEVEN  ******
C             MATERIALS AND TWO COORDINATE SYSTEMS
C
C              MATERIAL DESIGNATORS ARE
C
C                A - NITROGEN
C                B - OXYGEN
C                C - CARBON MONOXIDE
C                D - CARBON DIOXIDE
C                E - WATER
C                F - HYDROGEN
C                G - CARBON(SOLID)
C
C
C
      COMMON Z(83000)
      CALL INPUT
   10 CALL CDT
      CALL EDIT
      IF(SENSE LIGHT 1) 50,20
   20 CALL PH1
      CALL PH2
      CALL INTFACE
      CALL POC
      GO TO 10
   50 CALL PRINT2
      STOP
      END
```

```
      SUBROUTINE INPUT
C
C
C                  D  I  M  E  N  S  I  O  N
C
      DIMENSION MX(6200),AMY(6200),P(6200),J(6200),V(6200),
     1AMA(6200),AM1(6200),AMF(6200),A11(6200),AME(6200),AMF(6200),
     2AMG(6200),THETA(6200),DY(200),XX(101),YY(201)
      DIMENSION X(100),    Y(200),    Z(150),    I2(150),   FLEFT(200)
      DIMENSION TAU(100),  PL(200),   UL(200),   P2(200),   JR(200)
      DIMENSION IW1(15),   YAMO(200), SIGO(200), GAMO(200), DX(100)
```

```
COMMON           Z       ,XX      ,UR      ,PR      ,YY
COMMON           AID     ,AIX     ,AM      ,THETA   ,AMX     ,AREA
COMMON           BIG     ,BOUNCE  ,DDXN    ,DDVK    ,DKE     ,DVK
COMMON           DX      ,DY      ,E       ,FD      ,FS      ,FX
COMMON           OUT     ,P       ,PABOVE  ,PBLO    ,PIDTS   ,PPABOV
COMMON           PRR     ,PUL     ,QDT     ,RC      ,REZ     ,RHO
COMMON           RL,RR,SIG,QDDFL,SWITCH ,TABLM
COMMON           TAU     ,TAUDTS  ,TAUDTX  ,U       ,UK      ,URR
COMMON           UT      ,UU      ,UUU     ,UTEF    ,UVMAX   ,V
COMMON           VABOVE  ,VBLO    ,VEL     ,VK      ,VT      ,VTEF
COMMON           VV      ,VVABOV  ,VVBLO   ,W2      ,W3      ,WPS
COMMON           WS      ,WSA     ,WSB     ,WSC     ,XL      ,XLF
COMMON           XN      ,XR      ,Y_      ,Y_W     ,YN      ,YU
COMMON           ZMAX    ,I       ,II      ,IN      ,IR      ,IWS
COMMON           IWSA    ,IWSB    ,IWSC    ,IW1     ,J       ,JN
COMMON           JP      ,JP      ,K       ,KN      ,KP      ,KR
COMMON           KRM     ,L       ,M       ,MA      ,MB      ,MC
COMMON           MD      ,ME      ,MZ      ,N       ,NK      ,NKMAX
COMMON           NK1     ,NO      ,NR      ,IW2
COMMON  AMA,AMB,AMC,AMD,AME,AMF,AMG
```

E Q U I V A L E N C E

```
DEQUIVALENCE     (Z,IZ,PROB),     (Z(2),CYCLE),    (Z(3),DT),
1(Z(4),PRINTS),  (Z(5),PRINTL),   (Z(6),DUMPTZ),   (Z(7),CSTOP),
2(Z(8),PID),     (Z(9),TMZ),      (Z(10),GAM),     (Z(11),GAMD),
3(Z(12),GAMX),   (Z(13),ETH),     (Z(14),FFA),     (Z(15),FFB),
4(Z(16),TMDZ),   (Z(17),TMXZ),    (Z(18),XMAX),    (Z(19),TXMAX),
5(Z(20),TYMAX),  (Z(21),AMDM),    (Z(22),AMXM),    (Z(23),DNN),
6(Z(24),DMIN),   (Z(25),FEF),     (Z(26),DTNM),    (Z(27),CVIS),
7(Z(28),NPR),    (Z(29),NPRI),    (Z(30),NC),      (Z(31),NPC),
8(Z(32),NRC),    (Z(33),IMAX),    (Z(34),IMAXA),   (Z(35),JMAX),
9(Z(36),JMAXA),  (Z(37),KMAX),    (Z(38),KMAXA),   (Z(39),NMAX)
DEQUIVALENCE     (Z(40),ND),      (Z(41),KDT),     (Z(42),IXMAX),
1(Z(43),NOD),    (Z(44),NOPR),    (Z(45),NIMAX),   (Z(46),NJMAX),
2(Z(47),I1),     (Z(48),I2),      (Z(49),I3),      (Z(50),I4),
3(Z(51),N1),     (Z(52),N2),      (Z(53),N3),      (Z(54),N4),
4(Z(55),N5),     (Z(56),N6),      (Z(57),N7),      (Z(58),N8),
5(Z(59),N9),     (Z(60),N10),     (Z(61),N11),     (Z(62),NRM),
6(Z(63),TRAD),   (Z(64),XNRG),    (Z(65),SN),      (Z(66),DXN),
7(Z(67),RADER),  (Z(68),RADET),   (Z(69),RADEB),   (Z(70),DTRAD),
8(Z(71),REZFCT), (Z(72),RSTOP),   (Z(73),SHELL),   (Z(74),BROUND),
9(Z(75),TOZONE), (Z(76),ECK),     (Z(77),SBOUND),  (Z(78),X1)
DEQUIVALENCE     (Z(79),X2),      (Z(80),Y1),      (Z(81),Y2),
1(Z(82),CABLN),  (Z(83),VISC),    (Z(84),T),       (Z(85),GMAX),

2(Z(86),WSG),    (Z(87),WSGX),    (Z(88),GMADR),   (Z(89),GMAXP),
3(Z(90),S1),     (Z(91),S2),      (Z(92),S3),      (Z(93),S4),
4(Z(94),S5),     (Z(95),S6),      (Z(96),S7),      (Z(97),S8),
5(Z(98),S9),     (Z(99),S10),
7(Z(130),NFITA), (Z(131),NFITT),  (Z(132),NFITR),
8(Z(133),NPARA), (Z(134),NPART),  (Z(135),NPARR)
```

```
C
      EQUIVALENCE(P(2,1),GAM),(P(4,1),AMC)
      EQUIVALENCE       (UR,UL,FLEFT),    (PR,PL,SIGC)
      EQUIVALENCE       (XX(2),X(1)),     (YY(2),Y(1))
C
C
C
      SENSE LIGHT 3
      PRINT 3013
 3013 FORMAT(*1*)
C
      READ  (5,9004)IWS
      WRITE (6,9004)IWS
    9 CALL CARDS
C
C                 READ TAPE
C
C
C
C
C
 1001 NZ=150
      IWS=0
 1003 REWIND 4
 1004 READ(4)PR(1),PR(2),NR
      NR=NR+5
 1005 IF(PR(1)-555.0)1011,1016,1011
 1011 IF(PR(1).EQ.555.)1012,1013
 1012 DO 1014 I=1,6
 1014 BACKSPACE 4
      GO TO 1023
 1013 IWS=IWS+1
 1011 IF(XMOD(IWS,3))9002,9002,1013
 1015 IF(Z(235)-PR(2))1023,1023,1022
 1021 READ(4) DUM
      READ(4) DUM
      READ(4) DUM
      READ(4) DUM
      READ(4) DUM
      GO TO 1004
 1023 READ(4)(Z(I),I=1,12)
      IF(ABSF(PR(2)-Z(235))-.1)1024,1024,9001
 1024 READ(4) (U(I),V(I),AMX(I),ALX(I),P(I),THETA(I),I=1,KMAXA)
      READ(4)(AM1(I),AM2(I),AMC(I),AM3(I),AME(I),AMF(I),AM5(I),
     1I=1,KMAXA)
      READ(4)X(0),(X(I),TAU(I),I=1,IMAX)
      X(0) = 0.0
      READ(4)(Y(I),I=0,JMAX)
 1034 READ(4)FS,RC,RP
 1035 IF(FS-555.0)9004,1041,1039
 1039 IF(RC-666.0)9005,1041,9006
C
C    ******* END OF READ TAPE ********************************
C
C         READ IN REMAINING INPUT CARDS
C
```

```
1040  CALL CARDS
      IF (NC) 40,42,41
   40 DO 45 K=1,KMAX1
   45 P(K)=0.0
      T=T-DTMA
      NC=NC-1
      CYCLE=NC
      NPC=NPC-1
      UVMAX=1.0
   41 DO 50 I=1,IMAX
   50 DX(I)=X(I)-X(I-1)
      DO 55 J=1,JMAX
   55 DY(J)=Y(J)-Y(J-1)
      JX=IZ-4
   60 DO 70 I=1,JX,4
      K=I+3
      DO 65 J=I,K
      IF(Z(J))70,65,70
   65 CONTINUE
      GO TO 42
   70 CONTINUE
      WRITE (6,9111) I,(Z(L),L=I,K)
      WRITE (6,9112) I,(Z(L),L=I,K)
   80 CONTINUE
      CALL PRINT2
      Z(1,4) = 1.1334/1.7
      GO TO 1(80)
C
C                 READ IN REMAINING INPUT CARDS
C
C
C
C
C                 ERROR
 9901 NK=1023
      GO TO 9999
 9902 NK=1011
      GO TO 9999
 9903 NK=1036
      GO TO 9999
 9904 NK=1031
      GO TO 9999
 9905 NK=2003
      GO TO 9999
 9907 NK=2012
 9999 NR=1
      CALL ERROR(NR,NK)
C
10000 RETURN
C
C                 FORMATS
 9001 FORMAT(7F11.3,I2)
 9004 FORMAT(I1,'1H
     1                    )
 9111 FORMAT (I4,4(4X,1PE16.7))
 9112 FORMAT(I4,'029)
C
      END
```

NOT REPRODUCIBLE

```
      SUBROUTINE CARDS
      DIMENSION TABLE(1),CARD(7),LABLE(1)
      COMMON           TABLE
      EQUIVALENCE(TABLE(1),LABLE(1))
      WRITE (6,10)
    1 READ  (5,11)IEND,LOC,NUMWPC,(CARD(I),I=1,NUMWPC)
      WRITE (6,12)IEND,LOC,NUMWPC,(CARD(I),I=1,NUMWPC)
      DO 4 I=1,NUMWPC
      J=LOC+I-1
      IF(IEND-2) 2,5,?
    5 LABLE(J)=IFIX(CARD(I))
      GO TO 4
    2 TABLE(J)=CARD(I)
    4 CONTINUE
      IF(IEND-1)1,3,1
    3 RETURN
C                FORMATS
   10 FORMAT(20H1SH LL    INPUT CARDS///)
   11 FORMAT(I1,I5,I1,0P7E3.4)
   12 FORMAT(1H I4,I7,I3,1P7E14.5)
      END



      SUBROUTINE PRINTZ
      COMMON Z(150)
      PRINT 8112,(Z(I),I=1,64)
      PRINT 8113,(Z(I),I=65,104)
      PRINT 8114,(Z(I),I=105,150)
 8112 FORMAT(4H1   1,3X,6HMPROB  E16.8,5X,6HCYCLE E16.8,5X,6HDT     E16.8,5
     AX,6HPRINTS E16.8/4H    5,3X,6HPRINTL E16.8,5X,6HJUMPT7E16.8,5X,6HCSTO
     8P E16.8,5X,6HPIO/  E16.8/4H    9,3X,6HT47   E16.8,5X,6HGAM   E16.8,
     C5X,6HGAMD   ,E16.8,5X,6HGAMX  E16.8/4H   13,3X,6HETH   E16.8,5X,6HFF
     DA   E16.8,5X,6HFF3   E16.8,5X,6HTMTZ  E16.8/4H   17,3X,6HTMXZ  E16.
     E8,5X,6HXMAX  E16.8,5X,6HTXMAX E16.8/4H   21,3X,6HA
     FMDM E16.8,5X,6HAMXM  E16.8,5X,6HDNII   E16.8,5X,6HDMIN  E16.8/4H
     G25,3X,6HFEF   E16.8,5X,6HDT1A  E16.8,5X,6HCVIS  E16.8,5X,6HNPR   1
     HX,I15/4H   29,3X,7HNPRI   I15,5X,7HNC      I15,5X,7HNPC     I15,5X,7H
     INRC    I15/4H   33,3X,7HIMAX   I15,5X,7HIMAXA  I15,5X,7HJMAX   I15,
     J5X,7HJMAXA  I15/4H   37,3X,7HKMAX   I15,5X,7HKMAXA  I15,5X,7HNMAX
     K I15,5X,7HND      I15/4H   41,3X,7HKDT     I15,5X,7HIXMAX  I15,5X,7HN
     LOD    I15,5X,7HNDPR   I15/4H   45,3X,7HNIMAX  I15,5X,7HNJMAX  I15,5
     MX,7HI1     I15,5X,7HI2     I15/4H   49,3X,7HI3     I15,5X,7HI4
     N115,5X,7HN1     I15,5X,7HN2     I15/4H   53,3X,7HN3     I15,5X,7HN4
     O     I15,5X,7HN5     I15,5X,7HN6     I15/4H   57,3X,7HN7     I15,5X
     P,7HN8     I15,5X,7HN9     I15,5X,7HN10    I15/4H   61,3X,7HN11    I
     Q15,5X,7HNP4    I15,5X,6HTRAD  E16.8,5X,6HXNPG  E16.8/4H   65,3X,6HS
     RN    E16.8,5X,6HDX*I   E16.8,5X,6HRADCP E16.8,5X,6HPADET E16.8)
 8113 FORMAT(4H   69,3X,6HRADCB E16.8,5X,6HTRAD E16.8,5X,6HREZFCTE16.8,5
     AX,6HRSTOP  E16.8/4H   73,3X,6HSHELL E16.8,5X,6HBOUNDE16.8,5X,6HTCZO
     8NE E16.8,5X,6HSCK   E16.8/4H   77,3X,6HSBOUNDE16.8,5X,6HX1     E16.8,
     C5X,6HX2     E16.8,5X,6HY1     E16.8/4H   81,3X,6HY2     E16.8,5X,6HCAB
     OLN E16.8,5X,6HVISC  E16.8,5X,6HT      E16.8/4H   85,3X,6HGMAX  E16.8
     E,5X,6HMSGO  E16.8,5X,6HMSGX  E16.8,5X,6HGMADP E16.8/4H   89,3X,6HGM
     FAXR E16.8,5X,6HS1     E16.8,5X,6HS2     E16.8,5X,6HS3     E16.8/4H   9
     G3,3X,6HS4     E16.8,5X,6HS5     E16.8,5X,6HS6     E16.8,5X,6HS7     E1
     H6.8/4H   97,3X,6HS8     E16.8,5X,6HS9     E16.8,5X,6HS10    E16.8,5X,6
```

```
      IMZ(100)E16.8/4H 101,9X,E16.8,3(11X,E16.8)
  811 FORMAT(* 105*9XE16.8,3(11XE16.8)/* 109*9XE16.8,3(11XE16.8)/* 113*9
     AXE16.8,3(11XE16.8)/* 117*9XE16.8,3(11XE16.8)/* 121*9XE16.8,3(11XE1
     86.8)/* 125*9XE16.8,3(11XE16.8)/* 129*9XE16.8,3(12XI15)/* 133*10XI1
     C5,2(12XI15),11XE16.8/* 137*9XE16.8,3(11XE16.8)/* 141*9XE16.8,3(11X
     DE16.8)/* 145*9XE16.8,3(11XE16.8)/* 149*9XE16.8,11XE16.8)
      RETURN
      END


      SUBROUTINE COT
C
C                D  I  M  E  N  S  I  O  N
C
      DIMENSION AIX(6200),AMX(6200),P(6200),U(6200),V(6200),
     1AMA(6200),AMB(6200),AMC(6200),AMD(6200),AME(6200),AMF(6200),
     2AMG(6200),THETA(6200),DY(200),XX(101),YY(201)
      DIMENSION X(100),    F(200),    Z(150),    IZ(150),   FLEFT(200)
      DIMENSION TAU(100),  PL(200),   UL(200),   PR(200),   UR(200)
      DIMENSION IW1(15),   YAMC(200), SIGC(200), GAMC(200), DX(100)

      COMMON           Z      ,XX     ,UR     ,PR     ,F
      COMMON           AID    ,AIX    ,AM     ,THETA  ,AMX    ,AREA
      COMMON           BIG    ,BOUNCE ,DDXN   ,DDVK   ,DKE    ,DVK
      COMMON           DX     ,DY     ,E      ,FD     ,FS     ,FX
      COMMON           OUT    ,P      ,PABOVE ,PBLO   ,PIDTS  ,PPABOV
      COMMON           PRR    ,PUL    ,RDT    ,RC     ,REZ    ,RHO
      COMMON           RL,RR,SIG,SOOOFL,SWITCH ,TABLM
      COMMON           TAU    ,TAUDTS ,TAUDTX ,U      ,UK     ,URR
      COMMON           UT     ,UU     ,UUU    ,UTEF   ,UVMAX  ,V
      COMMON           VABOVE ,VBLO   ,VEL    ,VK     ,VT     ,VTEF
      COMMON           VV     ,VVABOV ,VVBLO  ,W2     ,W3     ,WPS
      COMMON           WS     ,WSA    ,WSB    ,WSC    ,XL     ,XLF
      COMMON           XN     ,XR     ,YL     ,YLW    ,YN     ,YU
      COMMON           ZMAX   ,I      ,II     ,IN     ,IR     ,IWS
      COMMON           IWSA   ,IWSB   ,IWSC   ,IW1    ,J      ,JN
      COMMON           JP     ,JP     ,K      ,KN     ,KP     ,KR
      COMMON           KRM    ,L      ,M      ,MA     ,MB     ,MC
      COMMON           MD     ,ME     ,MZ     ,N      ,NK     ,NKMAX
      COMMON           NK1    ,NO     ,NR     ,NW2
      COMMON  AMA,AMB,AMC,AMD,AME,AMF,AMG
C
C
C                E  Q  U  I  V  A  L  E  N  C  E
C
      EQUIVALENCE    (Z,IZ,PROB),     (Z(2),CYCLE),     (Z(3),DT),
     1(Z(4),PRINTS),  (Z(5),PRINTL),   (Z(6),DUMPT7),    (Z(7),CSTOP),
     2(Z(8),PIDY),    (Z(9),TMZ),      (Z(10),GAM),      (Z(11),GAMD),
     3(Z(12),GAMX),   (Z(13),ETH),     (Z(14),FFA),      (Z(15),FFB),
     4(Z(16),TMDZ),   (Z(17),TMXZ),    (Z(18),XMAX),     (Z(19),TXMAX),
     5(Z(20),TMAX),   (Z(21),AMDM),    (Z(22),AMX4),     (Z(23),DNN),
     6(Z(24),DMIN),   (Z(25),FEF),     (Z(26),DTNA),     (Z(27),CVIS),
     7(Z(28),NPR),    (Z(29),NPRI),    (Z(30),NC),       (Z(31),NPC),
     8(Z(32),NRC),    (Z(33),IMAX),    (Z(34),IMAXA),    (Z(35),JMAX),
     9(Z(36),JMAXA),  (Z(37),KMAX),    (Z(38),KMAXA),    (Z(39),NMAX)
```

```
     0EQUIVALENCE      (Z(40),ND),       (Z(41),KDT),      (Z(42),IXMAX),
     1(Z(43),NOD),     (Z(44),NOPR),     (Z(45),NIMAX),    (Z(46),NJMAX),
     2(Z(47),I1),      (Z(48),I2),       (Z(49),I3),       (Z(50),I4),
     3(Z(51),N1),      (Z(52),N2),       (Z(53),N3),       (Z(54),N4),
     4(Z(55),N5),      (Z(56),N6),       (Z(57),N7),       (Z(58),N8),
     5(Z(59),N9),      (Z(60),N10),      (Z(61),N11),      (Z(62),NRM),
     6(Z(63),TRAD),    (Z(64),XNRG),     (Z(65),SN),       (Z(66),DXN),
     7(Z(67),RADER),   (Z(68),RADET),    (Z(69),RADEB),    (Z(70),DTRAD),
     8(Z(71),REZFCT),  (Z(72),RSTOP),    (Z(73),SMELL),    (Z(74),BBOUND),
     9(Z(75),TOZONE),  (Z(76),ECK),      (Z(77),SBOUND),   (Z(78),X1)
     0EQUIVALENCE      (Z(79),X2),       (Z(80),Y1),       (Z(81),Y2),
     1(Z(82),CABLN),   (Z(83),VISC),     (Z(84),T),        (Z(85),GMAX),
     2(Z(86),WSGD),    (Z(87),WSGA),     (Z(88),SMAJR),    (Z(89),SMAXR),
     3(Z(90),S1),      (Z(91),S2),       (Z(92),S3),       (Z(93),S4),
     4(Z(94),S5),      (Z(95),S6),       (Z(96),S7),       (Z(97),S8),
     5(Z(98),S9),      (Z(99),S10),
     7(Z(130),NFITB),  (Z(131),NFITT),   (Z(132),NFITR),
     8(Z(133),NPARB),  (Z(134),NPART),   (Z(135),NPARR)
C
      EQUIVALENCE(P(200),GAMC),(P(400),YAMC)
      EQUIVALENCE       (UR,UL,FLEFT),    (PR,PL,SIGC)
      EQUIVALENCE       (XX(2),X(1)),     (YY(2),Y(1))
      EQUIVALENCE (Z(105),RATA), (Z(106),RATB), (Z(107),RATC)
      EQUIVALENCE (Z(108),RATD), (Z(109),RATE), (Z(110),RATF)
      EQUIVALENCE (Z(111),RATG), (Z(112),ENIA), (Z(113),ENIB)
      EQUIVALENCE (Z(114),ENIC), (Z(115),ENID), (Z(116),ENIE)
      EQUIVALENCE (Z(117),ENIF), (Z(118),ENIG)
      EQUIVALENCE (Z(120),DELVEL), (Z(121),ENSUM), (Z(122),EXEN)
      EQUIVALENCE (Z(123),EXPHO), (Z(124),THICK), (Z(125),AJCEL)
      EQUIVALENCE (Z(126),TOTLEN)
C
C
C
C
      IF (SENSE LIGHT 3) 200,210
  200 SENSE LIGHT 3
      IF (NC) 21),205,205
  205 GO TO 3325
  210 JBOT = Z(144)
      JSTART=Z(147)
      IF (NC) 7,8,8
    7 Z(93) = AIX(10)
      Z(94) = AMX(10)
      Z(95) = AMA(10)
      Z(96) = AMB(10)
    8 DO 300 I=1,IMAX
      K2 = I + IMAX
      K3 = K2 + IMAX
      E02 = AIX(K2)+0.5*V(K2)*V(K2)*AMX(K2)
      E03 = AIX(K3)+0.5*V(K3)*V(K3)*AMX(K3)
      AMX(K2) = Z(94)
      AMX(K3) = Z(94)
      AMA(K2) = Z(95)
      AMA(K3) = Z(95)
      AMB(K2) = Z(96)
      AMB(K3) = Z(96)
      AIX(K2) = Z(93)
      AIX(K3) = Z(93)
```

```
      U(K2) = 0.J
      U(K3) = 0.0
      V(K2) = DETVEL
      V(K3) = DETVEL
      EN2 = AIX(K2)+J.5*V(K2)*V(K2)*AMX(K2)
      EN3 = AIX(K3)+0.5*V(K3)*V(<3)*AMX(K3)
  300 ETH = ETH + EN2 + EN3 - E02 - E03
      IF (NC-2) 1,2,2
    1 ENSUM = J.J
      TOTLEN = 0.0
      GO TO 3000
    2 DETLEN = DETVEL*DT
      DETMAS = DETLEN*THICK*EXPHO

      EKENVO = 0.5*DETMAS*(DETVEL**2)
      EKENUO = 0.0
      JCELL = AJCEL
      KCELL = 2 + (JCELL - 1)*IMAX
      EKENUO = 0.5*AMX(KCELL)*(U(KCELL)**2)
      EKENVO = 0.5*AMX(KCELL)*(V(KCELL)**2)
      U(KCELL)=(U(KCELL)*AMX(KCELL))/(AMX(KCELL)+DETMAS)
      V(KCELL)=(AMX(KCELL)*V(KCELL)+DETMAS*DETVEL)/(AMX(KCELL)+DETMAS)
      AMX(KCELL) = AMX(KCELL) + DETMAS
      EKENUN = 0.5*AMX(KCELL)*(U(KCELL)**2)
      EKENVN = 0.5*AMX(KCELL)*(V(KCELL)**2)
      DEKEN = EKENUO+EKENVO+EKENUO+EKENVO-EKENUN-EKENVN
      IF (DEKEN.LT.0.J) GO TO 9904
      AIX(KCELL) = AIX(KCELL) + DEKEN
      DMASA = RATA*DETMAS
      DMASB = RATB*DETMAS
      DMASC = RATC*DETMAS
      DMASD = RATD*DETMAS
      DMASE = RATE*DETMAS
      DMASF = RATF*DETMAS
      DMASG = RATG*DETMAS
      AMA(KCELL) = AMA(KCELL) + DMASA
      AMB(KCELL) = AMB(KCELL) + DMASB
      AMC(KCELL) = AMC(KCELL) + DMASC
      AMD(KCELL) = AMD(KCELL) + DMASD
      AME(KCELL) = AME(KCELL) + DMASE
      AMF(KCELL) = AMF(KCELL) + DMASF
      AMG(KCELL) = AMG(KCELL) + DMASG
      ENA = ENIA*DMASA
      ENB = ENIB*DMASB
      ENC = ENIC*DMASC
      END = ENID*DMASD
      ENE = ENIE*DMASE
      ENF = ENIF*DMASF
      ENG = ENIG*DMASG
      ENI = ENA+ENB+ENC+END+ENE+ENF+ENG
      AIX(KCELL) = AIX(KCELL) + ENI
      ENDEP = DETLEN*THICK*EXPHO*EXEN
      AIX(KCELL) = AIX(KCELL) + ENDEP
      ENSUM = ENSUM+ENDEP
      TOTLEN = TOTLEN+DETLEN
      WRITE (6,8444) DETLEN,TOTLEN
      WRITE (6,8445) ENDEP, ENSUM
```

```
8444 FORMAT(22H)DETONATION LENGTH IS,1PE12.3,5X,
    1  27HTOTAL DETONATION LENGTH IS  ,1PE12.3)
8445 FORMAT (21HOENERGY DEPOSITED IS ,1PE12.3,5X,
    1  26HTOTAL ENERGY DEPOSITED IS ,1PE12.3,/)
     EKN = EKENUD + EKENVD
     ETH = ETH + ENDEP + EKN + ENI
3000 VEL=0.0
     IF (SENSE LIGHT 3) 215,220
 215 SENSE LIGHT 3
     ININ = IMAX
     JNIN = JMAX
     GO TO 3005
 220 ININ = I1
     JNIN = I2
3005 DO 3050 I=1,ININ

3010 K=I+1
3015 DO 3050 J=1,JNIN
3020 IF(AMX(K).EQ.0..O-.AIX(K).EQ.0.) 3021,3025
3021 THETA(K)=0.
     P(K)=0.
     GO TO 3050
C
3025 IF(THETA(K).EQ.0.) THETA(K)=90.
     CALL STATE7
C
3030 IF(ABS(P(K))-1.0E-20)3035,3035,3040
3035 P(K)=0.0
3040 IF(WSGX-VEL)3050,3050,7045
3045 VEL=WSGX
3050 K=K+IMAX
 220 I=I+OTNA
3055 KOT=1
     UVMAX=-1.0
3070 DO 3255 I=1,I1
     K=I+1+IMAX*JJOT
3095 DO 3255 J=JSTART,I2
3100 KP=K+IMAX
     IF(AIX(K))4001,3255,4
   4 IF(AMX(K)/(TAU(I)*DY(J))-Z(138))3255,3255,3115
3115 SIG=DX(I)
3120 IF (DY(J)-SIG)3125,4000,4000
3125 SIG=DY(J)
4000 WS=SQRT(GMAX*TAU(I)*DY(J)*ABS(P(K))/(AMX(K)))
3205 WS=WS/SIG
3210 IF(UVMAX-WS)3215,3,3
3215 N10=I
     N11=J
     UVMAX=WS
   3 IF (GAM) 5,6,5
   6 WS=ABS(U(K))/TAU(I)*X(I)/.5*PIDY
     GO TO 3225
   5 WS = ABSF(U(K))/DX
3225 IF(UVMAX-WS)3230,3235,3235
3230 UVMAX=WS
     N10=I
     N11=J
```

```
 3230 WS=AMS(V(K))/DY(J)
 3240 IF(UVMAX-WS)3250,3250,3260
 3245 M10=I
      M11=J
      UVMAX=WS
 3250 CONTINUE
 3255 K=K+1MAX
      IF(UVMAX)9912,9912,90
    9 DT=.5/VEL/UVMAX*Z(13))
 3290 KOUT=0
  215 CONTINUE
    8 NC=NC+1
      CYCLE=NC
      NPC=NPC+1
 3300 IF(T)9904,3320,3310
 3310 IF(KOUT)9911,3315,3320
 3315 WRITE (6,3060)T,DTMA,DT
 3320 DTMA=DT
      GO TO 3325
C                  NEGATIVE MASS
 9901 NK=3020
      GO TO 9999
 9904 NK = 2
      GO TO 9999
 9906 NK=3305
      GO TO 9999
 9910 NK=3310
      GO TO 9999
C     THE DT WILL BE 0. OR NEGATIVE ,STOP
 9912 NK=1
      GO TO 9999
 9911 NK=85
 9999 NR=2
      CALL ERRER(NP,NK)
 3325 RETURN
 8000FORMAT (17H0CHANGE DT ... T=1PE12.3,11H      DT(N)=1PE12.3,
     1  13H      DT(N+1)=1PE12.3)
      END
```

```
      SUBROUTINE PH1
C                  D   I   M   E   N   S   I   O   N
C
      DIMENSION AIX(6200),AMX(6200),P(6200),U(6200),V(6200),
     1AMA(6200),AMB(6200),AMC(6200),AMD(6200),AME(6200),AMF(6200),
     2AMG(6200),THETA(6200),DY(200),XX(101),YY(201)
      DIMENSION X(100),    Y(200),    Z(150),    IZ(150),   FLEFT(200)
      DIMENSION TAU(100),  PL(200),   UL(200),   PR(200),   UR(200)
      DIMENSION IW1(15),   YAMC(200), SIGC(200), GAMC(200), DX(100)
```

```
      DIMENSION QL(200)
      COMMON           Z      ,XX     ,UR     ,PR     ,ff
      COMMON           AIO    ,AIX    ,AM     ,THETA  ,AMX    ,AREA
      COMMON           BIG    ,BOUNCE ,DDXN   ,DDVK   ,DKE    ,DVK
      COMMON           DX     ,DY     ,E      ,FD     ,FS     ,FX
      COMMON           OUT    ,P      ,PABOVE ,PBLO   ,PIDTS  ,PPABOV
      COMMON           PRR    ,PUL    ,QDT    ,RC     ,REZ    ,RHO
      COMMON           RL,RR,SIG,QDDDFL,SWITCH ,TABLM
      COMMON           TAU    ,TAUDTS ,TAUDTX ,U      ,UK     ,URR
      COMMON           UT     ,UU     ,UUU    ,UTEF   ,UVMAX  ,V
      COMMON           VABOVE ,VBLO   ,VEL    ,VK     ,VT     ,VTEF
      COMMON           VV     ,VVABOV ,VVBLO  ,W2     ,W3     ,WPS
      COMMON           WS     ,WSA    ,WSB    ,WSC    ,XL     ,XLF
      COMMON           XN     ,XR     ,fL     ,fLW    ,fN     ,fU
      COMMON           ZMAX   ,I      ,II     ,IN     ,IR     ,IWS
      COMMON           IWSA   ,IWSB   ,IWSC   ,IW1    ,J      ,JN
      COMMON           JP     ,JR     ,K      ,KN     ,KP     ,KR
      COMMON           KRM    ,L      ,M      ,MA     ,MB     ,MC
      COMMON           MD     ,ME     ,MZ     ,N      ,NK     ,NKMAX
      COMMON           NK1    ,NO     ,NR     ,IW2
      COMMON  AMA,AMB,AMC,AMD,AME,AMF,AMG
C
C
C                E  Q  U  I  V  A  L  E  N  C  E
C
     0EQUIVALENCE      (Z,IZ,PPO3),      (Z(2),CYCLE),     (Z(3),DT),
     1(Z(4),PRINTS),   (Z(5),PRINTL),    (Z(6),DUMPT7),    (Z(7),CSTOP),
     2(Z(8),PIDY),     (Z(9),TMZ),       (Z(10),GAM),      (Z(11),GAMD),
     3(Z(12),GAMX),    (Z(13),ETH),      (Z(14),FFA),      (Z(15),FF3),
     4(Z(16),TMDZ),    (Z(17),TMXZ),     (Z(18),XMAX),     (Z(19),TXMAX),
     5(Z(20),TYMAX),   (Z(21),AMDM),     (Z(22),AMXM),     (Z(23),DNN),
     6(Z(24),DMIN),    (Z(25),FEF),      (Z(26),DTNA),     (Z(27),CVIS),
     7(Z(28),NPR),     (Z(29),NPRI),     (Z(30),NC),       (Z(31),NPC),
     8(Z(32),NRC),     (Z(33),IMAX),     (Z(34),IMAXA),    (Z(35),JMAX),
     9(Z(36),JMAXA),   (Z(37),KMAX),     (Z(38),KMAXA),    (Z(39),NMAX)
     0EQUIVALENCE      (Z(40),ND),       (Z(41),KDT),      (Z(42),IXMAX),
     1(Z(43),NOD),     (Z(44),NOPR),     (Z(45),NIMAX),    (Z(46),NJMAX),
     2(Z(47),I1),      (Z(48),I2),       (Z(49),I3),       (Z(50),I4),
     3(Z(51),N1),      (Z(52),N2),       (Z(53),N3),       (Z(54),N4),
     4(Z(55),N5),      (Z(56),N6),       (Z(57),N7),       (Z(58),N8),
     5(Z(59),N9),      (Z(60),N10),      (Z(61),N11),      (Z(62),NRM),
     6(Z(63),TRAD),    (Z(64),XNRG),     (Z(65),SN),       (Z(66),DXN),
     7(Z(67),RADER),   (Z(68),RADET),    (Z(69),RADEB),    (Z(70),DTRAD),
     8(Z(71),REZFCT),  (Z(72),RSTOP),    (Z(73),SHELL),    (Z(74),BBOUND),
     9(Z(75),TOZONE),  (Z(76),ECK),      (Z(77),SBOUND),   (Z(78),X1)
     0EQUIVALENCE      (Z(79),X2),       (Z(80),f1),       (Z(81),f2),
     1(Z(82),CABLN),   (Z(83),VISC),     (Z(84),TT),       (Z(85),GMAX),
     2(Z(86),WSGD),    (Z(87),WSGX),     (Z(88),GMAJR),    (Z(89),GMAXP),
     3(Z(90),S1),      (Z(91),S2),       (Z(92),S3),       (Z(93),S4),
     4(Z(94),S5),      (Z(95),S6),       (Z(96),S7),       (Z(97),S8),
     5(Z(98),S9),      (Z(99),S10),
     7(Z(130),NFITD),  (Z(131),NFITT),   (Z(132),NFITR),
     8(Z(133),NPARD),  (Z(134),NPART),   (Z(135),NPARP)
C
      EQUIVALENCE(P(200),GAMC),(P(400),YAMC)
      EQUIVALENCE       (UR,UL,FLEFT),    (PR,PL,SIGC)
      EQUIVALENCE       (XX(2),X(1)),     (ff(2),f(1))
C
```

```
C
C
C
      MIN = -2
      LOK = -2
      NRT=0
      NRC=0
      JSTART=Z(147)
      JBOT=Z(144)
      UU=1.E+15
      UT=0.0
 8004 VEL=1.0
 8001 IF (GAM) 9000,3301,9000
 9000 RC = 1.0
      RR = RC
      GO TO 3304
 3301 RC=DX(1)/2.0
      RR=(X(1)+X(2))/2.0
 3304 K=2
      DO 3302 J=1,JMAX
      PL(J)=P(K)
      UL(J)=0.0
      QL(J)=0.0
 3302 K=K+IMAX
      DO 3360 I=1,I1
      K=I+1+JBOT*IMAX
      IF(CVIS)7002,7003,7003
 7002 VBLO=V(K)
      PBLO=0.0
      GO TO 7004
 7003 VBLO=0.0
      PBLO=P(K)
 7004 TAUDTS=TAU(I)*DT
      QB=0.0
      DO 3349 J=JSTART,I2
      A=0.
      B=0.
      UQ=U(K)
      VQ=V(K)
      PIDTS=1.0/(PID*DT*D(J))
      IF (GAM) 9002,9004,9002
 9002 PIDTS = 2.0*PIDTS
 9004 N = K+IMAX
 3305 IF(AMX(K))9902,3341,3306
 3306 IF(IMAX-I)9903,3311,3310
 3310 IF(AMX(K+1))9904,3312,3314
 3311 PRR=P(K)
      QR=0.0

 3307 ETH=ETH-PRR*U(K)/PIDTS*RC
      GO TO 3313
 3312 PRR=0.0
      QR=0.0
 3313 URR=RC*U(K)
      GO TO 3316
 3314 PRR=(P(K)+P(K+1))/2.0
 3315 URR=(U(K)*RC+U(K+1)*RR)/2.0
      RHO=0.5*(AMX(K)/TAU(I)+AMX(K+1)/TAU(I+1))/DY(J)
      DUR=U(K)-U(K+1)
```

```
      IF(DUR)2003,10001,2005
 2003 IF(S3)2004,10011,2004
10001 QR=0.0
      GO TO 3316
 2004 UTEF=ABSF(0.5*(U(K)+U(K+1)))
      QR=-S3*RHO*UTEF*DUR
      GO TO 3316
 2005 IF(S3)2007,2006,2007
 2006 UTEF=0.
      GO TO 2008
 2007 UTEF=ABSF(0.5*(U(K)+U(K+1)))
 2008 CS=SQRTF(1.4*PRR/RHO)
      QR=RHO*DUR*(S1*CS+S2*S2*DUR+S3*UTEF)
 3316 IF(JMAX-J)9905,3318,3320
 3318 PABOVE=P(K)
      A=1.
      QT=0.0
 3319 ETH=ETH-PABOVE*V(K)/2.*TAUDTS
      GO TO 3323
 3320 IF(AMX(N))9906,3322,3324
 3322 PABOVE=0.0
      QT=0.0
 3323 VABOVE=V(K)
      GO TO 3328
 3324 PABOVE=(P(K)+P(N))/2.0
      RHO=2.0*(AMX(K)+AMX(N))/TAU(T)/D(J)
      DVZ=V(K)-V(N)
      IF(DVZ)2009,10003,2011
 2009 IF(S3)2010,10003,2010
10003 QT=0.0
      GO TO 7001
 2010 VTEF=ABSF(0.5*(V(K)+V(N)))
      QT=-S3*RHO*VTEF*DVZ
      GO TO 7001
 2011 IF(S3)2016,2012,2016
 2012 VTEF=0.
      GO TO 2014
 2016 VTEF=ABSF(0.5*(V(K)+V(N)))
 2014 CS=SQRTF(1.4*PABOVE/RHO)
      QT=RHO*DVZ*(S1*CS+S2*S2*DVZ+S3*VTEF)
 7001 IF(JSTART-J)3325,7005,9905
 7005 IF(CVIS) 7006,7006,7006
 7006 PBLO=P(K)
      ETH=ETH+PBLO*V(K)/2.0*TAUDTS
 7006 B=1
 3325 VABOVE=(V(K)+V(N))/2.0
 3328 IF(VEL) 9907,3404,3400
 3400 G=6.3567E+08

      G=Z(108)*G*G/((G+(D(J)-D(J)/2.)**2)
      IF(A+B) 9,9,10
    9 WS=(PBLO-PABOVE+QB-QT)*TAUDTS/AMX(K)-G*DT
      GO TO 11
   10 WS=(PBLO-PABOVE+QB-QT)*TAUDTS/AMX(K)-0.5*G*DT
   11 V(K)=V(K)+WS
```

```
      IF(ABSF(V(K))-1.E-01) 3401,3401,3402
3401  V(K)=0.0
3402  U(K)=U(K)+(PL(J)-PRR+QL(J)-QR)/(AMX(K))*RC/PIDTS*2.0
      IF(ABSF(U(K))-1.E-01)3403,3403,3404
3403  U(K)=0.0
3404  IF(I-I1)6016,6005,6005
6005  IF(U(K))6605,6606,6605
6605  NRC=1
6606  IF (V(K)-Z(120))6607,6004,6607
6607  NRC=1
6004  IF (AIX(K) - Z(149)) 6015,6016,6015
6015  NRC=1
6016  CONTINUE
      IF(VEL)9907,2001,2000
2000  VQ=(VQ+V(K))/2.0
      UQ=(UQ+U(K))/2.0
      WS=(P(K)/2.*(VBLO-VABOVE)+QB*(VBLO-VQ)+QT*(VQ-VABOVE))*TAUDTS
      WT=(P(K)/2.*(UL(J)-URR)+QL(J)*(UL(J)-RC*UQ)+QR*(RC*UQ-URR))/PIDTS*
     1 2.0
      GO TO 2002
2001  WS=(VBLO-VABOVE)*.5*TAUDTS*P(K)
      WT=P(K)*(UL(J)-URR)/PIDTS
2002  RHO=WS+WT
      WSX=AIX(K) + RHO
      NIT = 1
      IF (NC.EQ.NIM.AND.K.EQ.LOK) PRINT 8801,NC,K,NIT,AIX(K),WSX,RHO,
     1  WS,WT,VBLO,VABOVE,UL(J),URR
8801  FORMAT (2X,I4,1X,I5,1X,I2,8(1X,E12.4),/)
1000  IF (WSX) 9908,1001,1001
1001  AIX(K) = WSX
      GO TO 3342
3343  PRR=0.0
      QR=0.0
      QT=0.0
      URR=U(K+1)*RR
      PABOVE=0.0
      VABOVE=V(N)
3342  VBLO=VABOVE
      PL(J)=PRR
      UL(J)=URR
      QL(J)=QR
      QB=QT
      K=N
3343  PBLO=PABOVE
      LL=K-IMAX
      IF(U(LL))6000,6001,6000
6000  NRT=1
6001  IF (V(LL) - Z(120)) 6002,6003,6002
6002  NRT=1
6003  IF (AIX(LL) - Z(149)) 6017,6018,6017
6017  NRT=1
6018  CONTINUE
```

```
 3356 RR=RR
      IF (GAM) 3360,9007,3360
 9007 RR=(X(I+1)+X(I+2))/2.0
 3360 CONTINUE
 3361 IF(VEL)9911,7030,3363
 3363 VEL=0.0
      GO TO 9001
C                 ERROR
 9902 NK=3305
      GO TO 9999
 9903 NK=3306
      GO TO 9999
 9904 NK=3310
      GO TO 9999
 9905 NK=3316
      GO TO 9999
 9906 NK=3320
      GO TO 9999
 9907 NK=3328
      GO TO 9999
 9908 NK=1000
      PRINT 8800,I,J,AIX(K),WS,WT,DT
 8800 FORMAT(30H ****NEGATIVE ENERGY FOUND**** ,2I5,5E12.4)
      GO TO 9999
 9911 NK=3361
 9999 NR=3
      CALL ERPER(NP,K)
 7030 I1=I1+NRC
 6203 I2=I2+NPT
      IF(I1-IMAX)6100,6100,6200
 6200 I1=IMAX
 6100 IF(I2-JMAX)6201,6201,6202
 6202 I2=JMAX
 6201 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PH2
C                 D  I  M  E  N  S  I  O  N
C
      DIMENSION AIX(6200),AMX(6200),P(6200),U(6200),V(6200),
     1AMA(6200),AMB(6200),AMC(6200),AMD(6200),AME(6200),AMF(6200),
     2AMG(6200),THETA(6200),DY(200),XX(101),YY(201),GAMA(200),
     3GAMB(200),GAMC(200),GAMD(200),GAME(200),GAMF(200),GAMG(200)
      DIMENSION X(100),     Y(200),     Z(150),     IZ(150),    FLEFT(200)
      DIMENSION TAU(100),   PL(200),    UL(200),    PR(200),    JR(200)
      DIMENSION IW1(15),    AMC(200),   SIGC(200),  GAMC(200),  DX(100)
      COMMON            Z       ,XX       ,UR       ,PR       ,YY
      COMMON            AID     ,AIX      ,AM       ,THETA    ,AMX      ,AREA
      COMMON            BIG     ,BOUNCE   ,DDXN     ,DDVK     ,DKE      ,DVK
      COMMON            DX      ,DY       ,E        ,ED       ,ES       ,EX
      COMMON            OUT     ,P        ,PABOVE   ,PBLO     ,PIOTS    ,PPABOV
      COMMON            PRR     ,PUL      ,QDT      ,RC       ,REZ      ,RHO
      COMMON            RL,RR,SIG,QDOOFL,SWITCH ,TABLM
      COMMON            TAU     ,TAUDTS   ,TAUDTX   ,U        ,UK       ,URR
      COMMON            UT      ,UU       ,UUC      ,UTEF     ,UVMAX    ,V
      COMMON            VABOVE  ,VBLO     ,VEL      ,VK       ,VT       ,VTEF
      COMMON            VV      ,VVABOV   ,VVBLO    ,W2       ,W3       ,WPS
      COMMON            WS      ,WSA      ,WSB      ,WSC      ,XL       ,XLF
      COMMON            XN      ,XR       ,YL       ,YLW      ,YN       ,YU
      COMMON            ZMAX    ,I        ,II       ,IN       ,IR       ,IWS
      COMMON            IWSA    ,IWSB     ,IWSC     ,IW1      ,J        ,JN
      COMMON            JP      ,JR       ,K        ,KN       ,KP       ,KR
      COMMON            KRN     ,L        ,M        ,MA       ,MB       ,MC
      COMMON            MD      ,ME       ,MZ       ,N        ,NK       ,NKMAX
      COMMON            NK1     ,NO       ,NR       ,IW2
      COMMON  AMA,AMB,AMC,AMD,AME,AMF,AMG
C
C
C                 E  Q  U  I  V  A  L  E  N  C  E
C
      DEQUIVALENCE      (Z,IZ,PRO3),      (Z(2),CYCLE),     (Z(3),DT),
     1(Z(4),PRINTS),    (Z(5),PRINTL),    (Z(6),DUMPTZ),    (Z(7),CSTOP),
     2(Z(8),PIOY),      (Z(9),TMZ),       (Z(10),GAM),      (Z(11),GAMD),
     3(Z(12),GAMX),     (Z(13),ETH),      (Z(14),FFA),      (Z(15),FFB),
     4(Z(16),TMDZ),     (Z(17),TMXZ),     (Z(18),XMAX),     (Z(19),TXMAX),
     5(Z(20),TYMAX),    (Z(21),AMDM),     (Z(22),AMXM),     (Z(23),DNN),
     6(Z(24),DMIN),     (Z(25),FEF),      (Z(26),DTNA),     (Z(27),CVIS),
     7(Z(28),NPR),      (Z(29),NPRI),     (Z(30),NC),       (Z(31),NPC),
     8(Z(32),NRC),      (Z(33),IMAX),     (Z(34),IMAXA),    (Z(35),JMAX),
     9(Z(36),JMAXA),    (Z(37),KMAX),     (Z(38),KMAXA),    (Z(39),NMAX)
      DEQUIVALENCE      (Z(40),NO),       (Z(41),KDT),      (Z(42),IXMAX),
     1(Z(43),NOD),      (Z(44),NOPR),     (Z(45),NIMAXA),   (Z(46),NJMAX),
     2(Z(47),I1),       (Z(48),I2),       (Z(49),I3),       (Z(50),I4),
     3(Z(51),N1),       (Z(52),N2),       (Z(53),N3),       (Z(54),N4),
     4(Z(55),N5),       (Z(56),N6),       (Z(57),N7),       (Z(58),N8),
     5(Z(59),N9),       (Z(60),N10),      (Z(61),N11),      (Z(62),NRM),
     6(Z(63),TRAD),     (Z(64),XNRG),     (Z(65),SN),       (Z(66),DXN),
     7(Z(67),RADER),    (Z(68),RADET),    (Z(69),RADEB),    (Z(70),DTRAD),
     8(Z(71),REZFCT),   (Z(72),RSTOP),    (Z(73),SHELL),    (Z(74),BBOUND),
     9(Z(75),TOZONE),   (Z(76),ECK),      (Z(77),SBOUND),   (Z(78),X1)
      DEQUIVALENCE      (Z(79),X2),       (Z(80),Y1),       (Z(81),Y2),
     1(Z(82),CABLN),    (Z(83),VISC),     (Z(84),T),        (Z(85),GMAX),
     2(Z(86),WSGD),     (Z(87),WSGX),     (Z(88),SHADR),    (Z(89),GMAXR),
     3(Z(90),S1),       (Z(91),S2),       (Z(92),S3),       (Z(93),S4),
```

```
     4(Z(94),S5),      (Z(95),S6),      (Z(96),S7),      (Z(97),S8),
     5(Z(98),S9),      (Z(99),S10),
     7(Z(130),NFITQ),  (Z(131),NFITT),  (Z(132),NFITR),
     8(Z(133),NPARQ),  (Z(134),NPART),  (Z(135),NPARR)
C
      EQUIVALENCE(P(200),GAMC),(P(400),VAMC)
     1,(P(600),GAMA),(P(800),GAMB),(P(1000),GAMDD),(P(1200),GAME),
     2(P(1400),GAMCC),(P(1600),GAMF),(P(1800),GAMG)
      EQUIVALENCE      (UR,UL,FLEFT),   (PR,PL,SIGC)
      EQUIVALENCE      (XX(2),X(1)),    (YY(2),Y(1))
C
C
C
C
C
      MIN = -2
      LOK = -2
      JBOT=Z(144)
      JSTART=Z(147)
      Z(141)=0.
      Z(142)=0.
      Z(143)=0
      NRT=
      NRC=0
      PEZ=1.)
      SENSE LIGHT 0
      PIOTS=1.0/(PI)(*)T)
  101 DO 103 J=1,JMAX
  102 GAMC(J)=0.)
      FLEFT(J)=0.0
      GAMA(J)=0.
      GAMB(J)=0.
      GAMCC(J)=0.
      GAMDD(J)=0.
      GAME(J)=0.
      GAMF(J)=0.)
      GAMG(J)=0.)
      VAMC(J)=0.)
      SIGC(J)=0.)
  103 CONTINUE
  104 DO 547 I=1,I1
      J=JSTART
      K=I+1+JBOT*IMAX
   81 IF(AMX(K))900,97,81
   81 IF(V(K))82,97,97
   97 AMMV=0.0
      GO TO 98
   82 AMMY=AMX(K)*V(K)*DT/DY(J)
      WS=AMX(K)
      IF(GAMC(J))790),83,83
  790 WS=WS+GAMC(J)
   83 IF(AMMY+WS)84,85,85
   84 AMMY=-WS
   85 IF(CVIS)106,99,99
  106 AMMU=AMMY*U(K)
      AMMV=AMMY*V(K)
      RAT=AMMY/AMX(K)
      DMAB=AMA(K)*RAT
```

```
      OMBB=AMB(K)*RAT
      OMCB=AMC(K)*RAT
      OMDB=AMD(K)*RAT
      OMEB=AME(K)*RAT
      OMFB=AMF(K)*RAT
      OMGB=AMG(K)*RAT
      DELPB=AIX(K)/AMX(K)+(U(K)**2+V(K)**2)/2.
      WS=(U(K)**2+V(K)**2)/2.
      ETH=ETH+AMMY*(AIX(K)/AMX(K)+WS)
C     *****MASS HAS LEFT THE BOTTOM*****
   94 SEZ=1.
      Z(143)=1.
      GO TO 107
   98 AMMV=0.
   99 AMMY=0.0
      OMAB=0.
      OMBB=0.
      OMCB=0.
      OMDB=0
      OMEB=0
      OMFB=0.0
      OMGB=0.0
      AMMU=0.0
      DELPB=0.0
  107 DO 546 J=JSTART,12
  108 L=K+IMAX
      AREA=0.0
      IF (K.EQ. 752) LOK=K
      VEL= .0
      FS=0.0
  210 IF(JMAX-J)211,211,207
  211 VEL=1.0
      GO TO 208
  207 IF(AMX(L))215,215,214
  214 IF(AMX(K))216,216,213
  215 VABOVE=V(L)
      GO TO 212
  216 IF(AMX(K))205,205,204
  205 VABOVE=0.0
      GO TO 212
  208 VABOVE=V(K)
      GO TO 212
  204 VABOVE=(DY(J+1)*V(K)+DY(J)*V(L))/(DY(J)+DY(J+1))
  212 CONTINUE
      FS=0.0
  404 IF(IMAX-I)412,412,405
  405 IF(AMX(K+1))411,411,409
  409 IF(AMX(K))413,410,407
  413 URB=U(K+1)
      GO TO 408
  411 IF(AMX(K))403,403,406
  403 URB=0.0
      GO TO 408
  412 FS=1.0
  406 URB=U(K)
      GO TO 408
  407 URB=(DX(I+1)*U(K)+DX(I)*U(K+1))/(DX(I)+DX(I+1))
  408 CONTINUE
```

NOT REPRODUCIBLE

```
  109 IF(AREA)9901,301,547
  301 IF(VABOVE)300,304,302
  302 IF(AMX(K))9900,304,303
  303 M=K
      MM=L
      JJ=J
      GO TO 307
  304 AMPY=0.0
      DMAT=0
      DMBT=0
      DMCT=0
      DMDT=0
      DMET=0
      DMFT=0.0
      DMGT=0.0
  305 AMUT=0.0
      AMVT=0.0
      DELET=0.0
      GO TO 501
  300 IF(VEL)9901,305,304
  305 IF(AMX(L))3903,304,306
  306 M=L
      MM=K
      JJ=J+1
  307 IF(VEL)6130,6130,6140
 6130 WSA=0.5*(V(L)+V(K))
      WSB=(V(L)-V(K))*DT/D(J)
      IF(ABS(V(M))*DT.GT.0.5*DY(JJ))WSB=0.
      VABOVE=WSA/(1.+WSB)
 6140 AMPY=AMX(M)*VABOVE/DY(JJ)*DT
 6141 RAT = AMPY/AMX(M)
      DMAT=AMA(M)*RAT
      DMBT=AMB(M)*RAT
      DMCT=AMC(M)*RAT
      DMDT=AMD(M)*RAT
      DMET=AME(M)*RAT
      DMFT=AMF(M)*RAT
      DMGT=AMG(M)*RAT
      IF(VEL.GT.0.) GO TO 501
      IF(AMA(M)/AMX(M).LT..001.AND.AMA(MM).LE.0.)DMAT=0.
      IF(AMB(M)/AMX(M).LT..001.AND.AMB(MM).LE.0.)DMBT=0.
      IF(AMC(M)/AMX(M).LT..001.AND.AMC(MM).LE.0.)DMCT=0.
      IF(AMD(M)/AMX(M).LT..001.AND.AMD(MM).LE.0.)DMDT=0.
      IF(AME(M)/AMX(M).LT..001.AND.AME(MM).LE.0.)DMET=0.
      IF(AMF(M)/AMX(M).LT..001.AND.AMF(MM).LE.0.)DMFT=0.
      IF(AMG(M)/AMX(M).LT..001.AND.AMG(MM).LE.0.)DMGT=0.
      ADELM=DMAT+DMBT+DMCT+DMDT+DMET+DMFT+DMGT
      ADELM=AMPY/ADELM
      DMAT=DMAT*ADELM
      DMBT=DMBT*ADELM
      DMCT=DMCT*ADELM
      DMDT=DMDT*ADELM
      DMET=DMET*ADELM
      DMFT=DMFT*ADELM
      DMGT = DMGT*ADELM
  501 IF(URR)500,504,502
  502 IF(AMX(K))3900,504,503
  503 M=K
```

```
      MM=K+1
      N=I
      GO TO 504
  504 AMMP=0.0
      DMAR=0
      DMBR=0
      DMCR=0
      DMDR=0
      DMER=0
      DMFR=0.0
      DMGR=0.0
      AMUR=0.0
      AMVR=0.0
      DELER=0.0
      GO TO 1
  505 IF(FS)9905,506,504
  506 IF(AMX(K+1))9904,504,507
  507 M=K+1
      MM=K
      N=I+1
  508 IF(FS)6100,6100,6110
 6100 WSA=0.5*(U(K)+U(K+1))
      WSB=(U(K+1)-U(K))*DT/DX(I)
      IF(ABS(U(M))*DT.GT.DX(M)*0.5)WSB=0.
      URR=WSA/(1.+WSB)
 6110 DEN=A14(M)/TAU(N)
      IF (GAM) 9991,9993,9992
 9992 CAPT = Y(2)**2.)
      GO TO 9994
 9993 CAPT = 1.0
 9994 AMMP = DEN/PIDTS*CAPT*URR
 6111 RAT=AMMP/A1X(M)
      DMAR=A1A(M)*RAT
      DMBR=A1B(M)*RAT
      DMCR=A1C(M)*RAT
      DMDR=A1D(M)*RAT
      DMER=A1E(M)*RAT
      DMFR=A1F(M)*RAT
      DMGR=A1G(M)*RAT
      IF(FS.GT.0.) GO TO 1
      IF(AMA(M)/AMX(1).LT..001.AND.A1A(MM).LE.0.)DMAR=0.
      IF(AMB(M)/AMX(1).LT..001.AND.A1B(MM).LE.0.)DMBR=0.
      IF(AMC(M)/AMX(1).LT..001.AND.A1C(MM).LE.0.)DMCR=0.
      IF(AMD(M)/AMX(1).LT..001.AND.A1D(MM).LT.0.)DMDR=0.
      IF(AME(M)/AMX(1).LT..001.AND.A1E(MM).LE.0.)DMER=0.
      IF(AMF(M)/AMX(1).LT..001.AND.A1F(MM).LE.0.)DMFR=0.
      IF(AMG(M)/AMX(1).LT..001.AND.A1G(MM).LE.0.)DMGR=0.
      ADELM=DMAR+DMBR+DMCR+DMDR+DMER+DMFR+DMGR
      ADELM=AMMP/ADELM
      DMAR=DMAR*ADELM
      DMBR=DMBR*ADELM
      DMCR=DMCR*ADELM
      DMDR=DMDR*ADELM
      DMER=DMER*ADELM
      DMFR=DMFR*ADELM
      DMGR=DMGR*ADELM
    1 IF (AMMP) 16,74,9920
 9920 IF (GAMC(J))74,74,9921
```

```
8821 IF (FS)6121,6120,74
6120 IF (AMX(K+1))9903,74,74
  74 JTAG=0
     IF(AMPY)3,4,4
     ITAG=1
     WSB=AMPY
     AMPY=0.0
     GO TO 64
     ITAG=0
  64 IF(AMMY)9,5,5
     IF(GAMC(J))7,6,5
     WS=AMX(K)
     GO TO 11
     WS=AMX(K)+GAMC(J)
     GO TO 11
     IF(GAMC(J))10,8,9
     WS=AMX(K)+AMMY
     GO TO 11
  10 WS=AMX(K)+GAMC(J)+AMMY
  11 WSA=AMPY+AMMP
  12 IF(WSA-WS)75,75,13
  13 RAT=WS/WSA
     IF((ITAG.EQ.1) GO TO 551
     AMPY=AMPY*RAT
     DMAT=DMAT*RAT
     DMBT=DMBT*RAT
     DMCT=DMCT*RAT
     DMDT=DMDT*RAT
     DMET=DMET*RAT
     DMFT=DMFT*RAT
     DMGT=DMGT*RAT
 551 IF(JTAG.EQ.1) GO TO 75
     AMMP=AMMP*RAT
     DMAR=DMAR*RAT
     DMBR=DMBR*RAT
     DMCR=DMCR*RAT
     DMDR=DMDR*RAT
     DMER=DMER*RAT
     DMFR=DMFR*RAT
     DMGR=DMGR*RAT
  75 IF(JTAG)14,73,14
  73 WSC=AMMP
  14 IF(ITAG)15,7011,15
  15 AMPY=WSB
     ITAG=0
     GO TO 60
  16 IF(FS)76,17,76
  76 WSC=AMMP
     GO TO 60
  17 IF(I+1-IMAX)18,18,9999
  18 UPPR=U(K+1)/2.0
     GO TO 20
  19 UPPR=(U(K+1)+U(K+2))/2.0
  20 IF(UPPR)39,39,21
  21 IF (SA1) 9997,9992,9997
9992 CART = X(I+1)**2.0
     GO TO 9994
9993 CART = 1.0
```

```
9994 URRR = URRR/TAU(I+1)*AMX(K+1)/PIDTS*CART
  22 IF(J-JSTART)9999,23,24
  23 VBLO=V(K+1)/2.0
     GO TO 26
  24 KP=K+1-IMAX
     VBLO=(V(K+1)+V(KP))/2.0
  26 IF(VBLO)25,38,38
  25 VBLO=AMX(K+1)/DY(J)*VBLO*DT
  27 IF(VEL)28,29,28
  28 VAB=V(K+1)/2.0
     GO TO 31
  29 KP=K+IMAX+1
     VAB=(V(K+1)+V(KP))/2.0
  31 IF(VAB)36,36,30
  30 VAB=AMX(K+1)/DY(J)*VAB*DT
  32 WS=AMX(K+1)
  33 WSA=URRR-AMMP-VBLO+VAB
  34 IF(WSA-WS)77,77,35
  35 AMMP=AMMP*WS/WSA
     RAT=WS/WSA
     DM4R=DM4R*RAT
     DMBR=DMBR*RAT
     DMCR=DMCR*RAT
     DMDR=DMDR*RAT
     DMER=DMER*RAT
     DMFR=DMFR*RAT
     DMGR=DMGR*RAT
  77 JTAG=1
     WSC=AMMP
     AMMP=0.0
     GO TO 2
  36 WS=AMX(K+1)
  37 WSA=URRR-AMMP-VBLO
     GO TO 34
  38 VBLO=0.0
     GO TO 27
  39 URRR=0.0
     GO TO 22
  40 IF(VEL)7000,41,7000
  41 IF(FS)42,43,42
  42 KP=K+IMAX
     URT=U(KP)/2.0
     GO TO 45
  43 KP=K+IMAX
     URT=(U(KP)+U(KP+1))/2.0
  45 IF(URT)46,46,70
  46 URT=0.0
     GO TO 47
  70 KP=K+IMAX
     IF (GAM)  9997,9996,9997
9996 CART = X(I)*2.0
     GO TO 9998
9997 CART = 1.0
9998 URT = URT/TAU(I)*AMX(KP)/PIDTS*CART
  47 IF(J+1-JMAX)48,49,9910
  48 KP=K+IMAX
     KL=KP+IMAX
     VABT=(V(KP)+V(KL))/2.0
```

```
      GO TO 51
   49 KP=K+IMAX
      KL=KP+IMAX
      VABT=V(KP)/2.0
   51 IF (VABT)8810,71,72
 8810 IF (AMX(K)) 9913,71,71
   71 VABT=0.0
      GO TO 60
   72 VABT=VABT*AMX(KP)/DY(J+1)*DT
   52 IF(GAMC(J+1)) 54,53,53
   53 WS=AMX(KP)
      GO TO 55
   54 WS=AMX(KP)+GAMC(J+1)
   55 WSA=VABT-AMPY+URT
      GO TO 57
   60 IF(GAMC(J+1))61,61,53
   61 WS=AMX(KP)+GAMC(J+1)
      GO TO 58
   59 WS=AMX(KP)
   58 WSA=-AMPY+URT
   57 IF(WSA-WS)7000,7000,56
   56 AMPY=AMPY*WS/WSA
      RAT=WS/WSA
      DMAT=DMAT*RAT
      DMBT=DMBT*RAT
      DMCT=DMCT*RAT
      DMDT=DMDT*RAT
      DMET=DMET*RAT
      DMFT=DMFT*RAT
      DMGT=DMGT*RAT
 7000 AMMP=WSC
  309 IF(AMPY)8834,8831,8833
 8833 IF(JMAX-J)3911,310,8832
 8832 KP=K+IMAX
 8835 IF(AMX(KP))8836,8837,310
 8837 IF(AMPY/(TAU(I)*DY(J+1))-TOZONE)8833,310,310
 8834 AMPY=0.0
      DMAT=0
      DMBT=0
      DMCT=0
      DMDT=0
      DMET=0
      DMFT=0.
      DMGT=0.
      GO TO 8831
 8834 CONTINUE
 8839 IF(AMX(K))8840,8842,325
 8840 IF(-AMPY/(TAU(I)*DY(J))-TOZONE)8841,325,325
 8841 AMPY=0.0
      DMAT=0
      DMBT=0
      DMCT=0
      DMDT=0
      DMET=0
      DMFT=0.
      DMGT=0.
      GO TO 8831
  311 DELH=GAMC(J)+AMPY-AMPY
```

```
  322 IF(VEL)9901,324,323
  323 WS=U(K)**2+V(K)**2
      ETH=ETH-AMPY*(AIX(K)/AMX(K)+WS/2.)
      AMUT=AMPY*U(K)
      AMVT=AMPY*V(K)
C     ****MASS HAS LEFT THE TOP
 6900 REZ=1.0
      Z(141)=1.
      GO TO 326
  324 AMUT=AMPY*U(K)
      AMVT=AMPY*V(K)
      GO TO 326
  325 CONTINUE
 8831 AMUT=AMPY*U(L)
      AMVT=AMPY*V(L)
      DELM=GAMC(J)-AMPY+AMMY
  326 IF(AMPY)327,328,328
  327 DELET=AIX(L)/AMX(L)+(U(L)**2+V(L)**2)/2.
      GO TO 333
  328 IF(AMMY)329,330,331
  329 DELET=DELEB
      GO TO 333
  330 IF(GAMC(J))331,332,332
  331 DELET=SIGC(J)
      GO TO 333
  332 IF(AMX(K).EQ.0.)GO TO 333
      DELET=AIX(K)/AMX(K)+(U(K)**2+V(K)**2)/2.
  333 SIGMU=FLEFT(J)+AMMU-AMUT
      SIGMV=YAMC(J)+AMMV-AMVT
      DELEK=GAMC(J)*SIGC(J)+AMMY*DELEB-AMPY*DELET
  509 IF(AMMP)8843,518,8844
 8844 IF(IMAX-I)9911,518,8845
 8845 IF(AMX(K+1))9900,8846,518
 8846 IF(AMMP/(TAU(I+1)*DY(J))-TOZONE)8847,518,518
 8847 AMMP=0.0
      DMAR=0
      DMBR=0
      DMCR=0
      DMDR=0
      DMER=0
      DMFR=0.
      DMGR=0.
      GO TO 518
 8843 IF(I-1)9911,512,8848
 8848 IF(AMX(K))8900,8849,512
 8849 IF(-AMMP/(TAU(I)*DY(J))-TOZONE)8850,512,512
 8850 AMMP=0.0
      DMAR=0
      DMBR=0
      DMCR=0
      DMDR=0
      DMER=0
      DMFR=0.
      DMGR=0.
      GO TO 518
  512 DELM=DELM-AMMP+AMX(K)
 8828 AMUR=AMMP*U(K+1)
      AMVR=AMMP*V(K+1)
```

```
      GO TO 525
  518 DELM=DELM-AMMP+AMX(K)
  521 CONTINUE
  522 IF(FS)9905,524,523
  523 WS=U(K)**2+V(K)**2
      IF(AMX(K).EQ.0.) GO TO 524
      ETH=ETH-AMMP*(AIX(K)/AMX(K)+WS/2.)
      AMUR=AMMP*U(K)
      AMVR=AMMP*V(K)
 6401 REZ=1.0
C    ****MASS HAS LEFT THE RIGHT****
      Z(142)=1.0
      GO TO 525
  524 AMUR=AMMP*U(K)
      AMVR=AMMP*V(K)
  525 SIGMU=SIGMU-AMUR
      SIGMV=SIGMV-AMVR
  526 TIC=0.0
  527 IF(AMMP)528,529,529
  528 DELER=AIX(K+1)/AMX(K+1)+(U(K+1)**2+V(K+1)**2)/2.
      GO TO 537
  529 IF(AMMY)530,531,531
  530 DELEP=DELEB
      GO TO 536
  531 IF(GAMC(J))532,533,533
  532 DELER=SIGC(J)
      GO TO 536
  533 IF(AMPY)535,535,534
  534 DELER=DELET
      GO TO 536
  535 IF(AMX(K).EQ.0.) GO TO 536
      DELER=AIX(K)/AMX(K)+(U(K)**2+V(K)**2)/2.
  536 TIC=1.0
  537 DELEK=DELEK-AMMP*DELER
      NIT = 2
      IF (NC.EQ.MIN.AND.K.EQ.LOK) PRINT 8005, NC,K,NIT,AMMP,DELER,
     1  GAMC(J),SIGC(J),AMMY,DELEB,AMPY,DELET
 8005 FORMAT (2X,I4,1X,I5,1X,I2,9(1X,E12.4),/)
  538 IF(TIC)9907,539,550
  550 WS=DELER
      GO TO 999
  539 WS=0.
      IF(AMX(K).GT.0.)WS=AIX(K)/AMX(K)+0.5*(U(K)**2+V(K)**2)
  999 IF(DELM)994,543,540
  994 IF(AMX(K)*1.E-6+DELM)9906,997,997
  997 DELM=0.0
      GO TO 543
  540 ENK=AMX(K)*WS+DELEK
  541 U(K)=(SIGMU+AMX(K)*U(K))/DELM
      IF(ABS(U(K))-1.E-2)1601,601,601
 1601 U(K)=0.
  601 V(K)=(SIGMV+AMX(K)*V(K))/DELM
      IF(ABS(V(K))-1.E-2)1701,1702,1702
 1701 V(K)=0.
 1702 CONTINUE
      IF(I-I1)603,6604,6604
 6604 IF(U(K)+V(K)-Z(120)+AIX(K)-Z(149))6605,6606,6605
 6605 NRC=1
```

```
6606  CONTINUE
 603  WS=U(K)**2+V(K)**2
 542  AIX(K)=ENK-DEL1*WS/2.
      NIT = 3
      IF (NC.EQ.MIN.AND.K.EQ.LOK) PRINT 6005, NC,K,NIT,AIX(K),ENK,DELM,
     1  WS,DELEK
 543  AMX(K)=DEL1
      AMA(K)=AMA(K)+DMAB-DMAT-DMAR+GAMA(J)
      AMB(K)=AMB(K)+DMBB-DMBT-DMBR+GAMB(J)
      AMC(K)=AMC(K)+DMCB-DMCT-DMCR+GAMCC(J)
      AMD(K)=AMD(K)+DMDB-DMDT-DMDR+GAMDD(J)
      AME(K)=AME(K)+DMEB-DMET-DMER+GAME(J)
      AMF(K)=AMF(K)+DMFB-DMFT-DMFR+GAMF(J)
      AMG(K)=AMG(K)+DMGB-DMGT-DMGR+GAMG(J)
      IF(AMX(K))9900,2007,544
2007  AIX(K)=0.0
      AMA(K)=0
      AMB(K)=0
      AMC(K)=0
      AMD(K)=0
      AME(K)=0
      AMF(K)=0.
      AMG(K)=0.
      U(K)=0.0
      V(K)=0.0
 544  GAMC(J)=AM1P
      GAMA(J)=DMAR
      GAMB(J)=DMBR
      GAMCC(J)=DMCR
      GAMDD(J)=DMDR
      GAME(J)=DMER
      GAMF(J)=DMFR
      GAMG(J)=DMGR
      FLEFT(J)=AMUR
      YAMC(J)=AMVR
      SIGC(J)=DELER
 545  AMMY=AMPY
      DMAB=DMAT
      DMBB=DMBT
      DMCB=DMCT
      DMDB=DMDT
      DMEB=DMET
      DMFB=DMFT
      DMGB=DMGT
      AMMU=AMUT
      AMMV=AMVT
      DELEB=DELET
 546  K=K+IMAX
      LL=K-IMAX
      IF(U(LL)+V(LL)-Z(120)+AIX(LL)-Z(149))6500,547,6500
6500  NRT=1
 547  CONTINUE
      I1=I1+NRC
      I2=I2+NRT
      IF(IMAX-I1)6700,5701,6702
6700  I1=IMAX
6701  CONTINUE
6702  IF(JMAX-I2)6800,6801,6802
```

```
6800 I2=JMAX
6801 CONTINUE
6802 GO TO 548
9901 NK=360
     GO TO 9999
9902 NK = 2004
     GO TO 9999
9903 NK=302
     GO TO 9999
9903 NK=305
     GO TO 9999
9904 NK=506
     GO TO 9999
9905 NK=500
     GO TO 9999
9906 NK=513
     GO TO 9999
9911 NK=8833
     GO TO 9999
9912 NK=105
     PRINT 8006,I,J,AMK(K),AMA(K),AMB(K),AMC(K),AMD(K),AME(K)
8006 FORMAT(2I5,6E14.4)
     GO TO 9999
9904 NK= 17
     GO TO 9999
9909 NK= 22
     GO TO 9999
9910 NK= 47
     GO TO 9999
9907 NK=539
9999 NP=4
     CALL ERPER(NP,NK)
 548 SUM=0.0
2005 LA=1HA
     LB=1HB
     LC=1HC
     LD=1HD
     LE=1HE
     LF=1HF
     LG=1HG
     DO 2001 I=1,I1
     K=I+1+JBOT*IMAX
     DO 2000 J=JSTART,I2
     IF(AMA(K).LT.0..AND.AMA(K).GT.-1.E-15) AMA(K)=0.
     IF(AMB(K).LT.0..AND.AMB(K).GT.-1.E-15) AMB(K)=0.
     IF(AMC(K).LT.0..AND.AMC(K).GT.-1.E-15) AMC(K)=0.
     IF(AMD(K).LT.0..AND.AMD(K).GT.-1.E-15) AMD(K)=0.
     IF(AME(K).LT.0..AND.AME(K).GT.-1.E-15) AME(K)=0.
     IF(AMF(K).LT.0..AND.AMF(K).GT.-1.E-15) AMF(K)=0.
     IF(AMG(K).LT.0..AND.AMG(K).GT.-1.E-15) AMG(K)=0.
     IF(AMA(K).LT.0)  PRINT 9201,I,J,LA,AMA(K)
     IF(AMB(K).LT.0)  PRINT 9201,I,J,LB,AMB(K)
     IF(AMC(K).LT.0)  PRINT 9201,I,J,LC,AMC(K)
     IF(AMD(K).LT.0)  PRINT 9201,I,J,LD,AMD(K)
     IF(AME(K).LT.0)  PRINT 9201,I,J,LE,AME(K)
     IF(AMF(K).LT.0) PRINT 9201,I,J,LF,AMF(K)
     IF(AMG(K).LT.0)  PRINT 9201,I,J,LG,AMG(K)
9201 FORMAT(26H NEGATIVE MASS FOUND AT I=,I5,2HJ=,I5,A1,15H MATERIAL MA
```

```
     1SS=,E19.9)
      TOT=AMA(K)+AMB(K)+AMC(K)+AMD(K)+AME(K)+AMF(K)+AMG(K)
      IF(TOT.EQ.0) GO TO 150
      ERR=TOT-AMX(K)
      F=1.-ERR/TOT
      AMA(K)=AMA(K)*F
      AMB(K)=AMB(K)*F
      AMC(K)=AMC(K)*F
      AMD(K)=AMD(K)*F
      AME(K)=AME(K)*F
      AMF(K)=AMF(K)*F
      AMG(K)=AMG(K)*F
  151 IF(AMX(K)) 2010,2001,2010
 2001 IF(AMX(K)/(TAU(I)*D(J))-1.E-5) 2010,2008,2004
 2004 WS=(U(K)**2+V(K)**2)/2.0
      Z(100)=Z(100)+AMX(K)
      WS=AIX(K)+AMX(K)*WS
      Z(111)=Z(111)+WS
      ETH=ETH-WS
      Z(102)=Z(102)+AMX(K)*U(K)
      Z(103)=Z(103)+AMX(K)*V(K)
      AMX(K)=0.0
      AIX(K)=0.0
      P(K)=0.0
      AMA(K)=0.
      AMB(K)=0.
      AMC(K)=0.
      AMD(K)=0.
      AME(K)=0.
      AMF(K)=0.
      AMG(K)=0.
      THETA(K)=0.
      U(K)=0.0
      V(K)=0.0
      GO TO 2000
 2008 IF(AIX(K)) 2004,2000,2000
 2004 SUM=SUM+AIX(K)
      AIX(K)=0.0
      PRINT 9200,I,J
 9200 FORMAT(20H ZERO ENERGY AT I,J=,2I5)
      GO TO 4002
 2000 K=K+IMAX
 2001 CONTINUE
 2003 ETH=ETH-SUM
      Z(104)=Z(104)+SUM
 8001 RETURN
      END
```

```
      SUBROUTINE EDIT
C                 D  I  M  E  N  S  I  O  N
C
      DIMENSION MSG(6)
      DIMENSION AIX(6200),AMX(6200),P(6200),U(6200),V(6200),
     1AMA(6200),AMB(6200),AMC(6200),AMD(6200),AME(6200),AMF(6200),
     2AMG(6200),THETA(6200),DY(200),XX(101),YY(201)
      DIMENSION X(100),     Y(200),     Z(150),     IZ(150),    FLEFT(200)
      DIMENSION TAU(100),   PL(200),    UL(200),    PR(200),    UR(200)
      DIMENSION IW1(15),    YAMC(200),  SIGC(200),  GAMC(200),  DX(100)

      COMMON            Z       ,XX       ,UR       ,PR       ,YY
      COMMON            AID     ,AIX      ,AM       ,THETA    ,AMX      ,AREA
      COMMON            BIG     ,BOUNCE   ,DDXN     ,DDVK     ,DKE      ,DVK
      COMMON            DX      ,DY       ,E        ,ED       ,ES       ,EX
      COMMON            OUT     ,P        ,PABOVE   ,PBLO     ,PIDTS    ,PPABOV
      COMMON            PRR     ,PUL      ,QDT      ,PC       ,REZ      ,RHO
      COMMON            RL,RR,SIG,TOLDEL,SWITCH ,TABLM
      COMMON            TAU     ,TAUDTS   ,TAUDTX   ,U        ,UK       ,URR
      COMMON            UT      ,UU       ,UUU      ,UTEF     ,UVMAX    ,V
      COMMON            VABOVE  ,VBLO     ,VEL      ,VK       ,VT       ,VTEF
      COMMON            VV      ,VVABOV   ,VVBLO    ,W2       ,W3       ,WPS
      COMMON            WS      ,WSA      ,WSB      ,WSC      ,XL       ,XLF
      COMMON            XN      ,XP       ,YL       ,YLW      ,YN       ,YU
      COMMON            ZMAX    ,I        ,II       ,IN       ,IR       ,IWS
      COMMON            IWSA    ,IWSB     ,IWSC     ,IW1      ,J        ,JN
      COMMON            JP      ,JR       ,K        ,KN       ,KP       ,KR
      COMMON            KRM     ,L        ,M        ,MA       ,MB       ,MC
      COMMON            MD      ,ME       ,MZ       ,N        ,NK       ,NKMAX
      COMMON            NK1     ,NO       ,NR       ,IW2
      COMMON   AMA,AMB,AMC,AMD,AME,AMF,AMG

C
C
C                 E  Q  U  I  V  A  L  E  N  C  E
C
      EQUIVALENCE       (Z,IZ,PROB),      (Z(2),CYCLE),     (Z(3),DT),
     1(Z(4),PPINTS),    (Z(5),PRINTL),    (Z(6),DUMPT7),    (Z(7),CSTOP),
     2(Z(8),PIDV),      (Z(9),TMZ),       (Z(10),GAM),      (Z(11),GAMD),
     3(Z(12),GAMX),     (Z(13),ETH),      (Z(14),FFA),      (Z(15),FFB),
     4(Z(16),TMDZ),     (Z(17),TMXZ),     (Z(18),XMAX),     (Z(19),TXMAX),
     5(Z(20),TYMAX),    (Z(21),AMDM),     (Z(22),AMXM),     (Z(23),DNN),
     6(Z(24),DMIN),     (Z(25),FEF),      (Z(26),DTNA),     (Z(27),CVIS),
     7(Z(28),NPR),      (Z(29),NPRI),     (Z(30),NC),       (Z(31),NPC),
     8(Z(32),NRC),      (Z(33),IMAX),     (Z(34),IMAXA),    (Z(35),JMAX),
     9(Z(36),JMAXA),    (Z(37),KMAX),     (Z(38),KMAXA),    (Z(39),NMAX)
      EQUIVALENCE       (Z(40),ND),       (Z(41),KDT),      (Z(42),IXMAX),
     1(Z(43),NOD),      (Z(44),NOPR),     (Z(45),NIMAX),    (Z(46),NJMAX),
     2(Z(47),I1),       (Z(48),I2),       (Z(49),I3),       (Z(50),I4),
     3(Z(51),N1),       (Z(52),N2),       (Z(53),N3),       (Z(54),N4),
     4(Z(55),N5),       (Z(56),N6),       (Z(57),N7),       (Z(58),N8),
     5(Z(59),N9),       (Z(60),N10),      (Z(61),N11),      (Z(62),NRM),
     6(Z(63),TRAD),     (Z(64),XNRG),     (Z(65),SN),       (Z(66),DXN),
     7(Z(67),RADER),    (Z(68),RADET),    (Z(69),RADEB),    (Z(70),DTRAD),
     8(Z(71),REZFCT),   (Z(72),RSTOP),    (Z(73),SHELL),    (Z(74),BBOUND),
     9(Z(75),TOZONE),   (Z(76),ECK),      (Z(77),SBOUND),   (Z(78),X1)
      EQUIVALENCE       (Z(79),X2),       (Z(80),Y1),       (Z(81),Y2),
     1(Z(82),CABLN),    (Z(83),VISC),     (Z(84),T),        (Z(85),GMAX),
     2(Z(86),WSGD),     (Z(87),WSGX),     (Z(88),GMAXR),    (Z(89),GMAXR),
```

```
     3(Z(90),S1),        (Z(91),S2),        (Z(92),S3),        (Z(93),S4),
     4(Z(94),S5),        (Z(95),S6),        (Z(96),S7),        (Z(97),S8),
     5(Z(98),S9),        (Z(99),S10),
     7(Z(130),NFITB),    (Z(131),NFITT),    (Z(132),NFITR),
     8(Z(133),NPARB),    (Z(134),NPART),    (Z(135),NPARR)
C
      EQUIVALENCE(P(201),GAMC),(P(401),AMC)
      EQUIVALENCE         (UP,UL,FLEFT),     (P2,PL,SIGC)
      EQUIVALENCE         (XX(2),A41),       (YY(2),Y(1))
C
C
C
C
  10? IF(SENSE SWITCH 4)122,104
  10- IF(SENSE LIGHT 3)106,1.8
  10) IF(CYCLE-CSTOP)112,122,122
  112 IF(MODE(CYCLE,DUMPT2))11-,1,11-
  11- IF(MODE(CYCLE,PRINTL))12(,12-,120
  12. IF(MODE(CYCLE,PRINTS))14C,6..C,14C
  14) IF(SENSE LIGHT 1)142,14-4
  142 REWIND  4
      SENSE LIGHT 1
      IF(GAMX)144,144,14.
  143 CALL STRADY
  14- GO TO 1C(00)
C
C
C         DUMP ON TAPE4
  12) SENSE LIGHT 1
      PRINTL=CYCLE
    1 IF(DUMPT7)30,3,3
    3 BACKSPACE  4
    - WS=5.5.C
      WRITE ( 4) WS,CYCLE,NP
   2) WRITE ( 4) (Z(L),L=1,1Z)
    ) WRITE(4)  (U(K),V(K),AMV(K),AMX(K),P(K),THETA(K),K=1,KMAXA)
      WRITE(4) (AMA(I),AMB(I),AMC(I),AMD(I),AME(I),
     1AMF(I),AMG(I),I=1,KMAXB)
    / WRITE ( 4) X(1),(Y(K),TAU(K),K=1,IMAX)
      WRITE ( 4) (Y(K),K=2,IMAX)
      WS=6.6.C
      WRITE ( 4) WS,WS,WS
      IF(SENSE SWITCH 2)25,24
   25 REWIND 4
      PRINT 8121,CYCLE
8121 FORMAT(*1*//////////10(10H**********)////* CYCLE*F5.0* BEGINS NEW T
     *APE *////10(10H**********)/*:*)
      PAUS  1.
      GO TO 4
   24 WRITE (6,8120) IC
   31 GO TO 114
C
C
  10) SENSE LIGHT 3
  12. SENSE LIGHT 4
6007 DO 6012 I=1,12
6012 PR(I)=3.C
      DO 6124 K=2,KMAX
```

NOT REPRODUCIBLE

```
      WSB=(U(K)**2+V(K)**2)/2.0
 6019 IF(AMX(K))9917,6028,6020
 6020 PR(5)=PR(5)+AIX(K)
      PR(6)=PR(6)+WSB*AMX(K)
      PR(8)=PR(8)+AMX(K)
 6028 CONTINUE
      PR(3)=PR(1)+PR(2)
      PR(7)=PR(5)+PR(6)
      AMX=PR(7)
      PR(9)=PR(1)+PR(5)
      PR(10)=PR(2)+PR(6)
      PR(11)=PR(3)+PR(7)
      PR(12)=PR(4)+PR(8)
      WSA=(ETH-PR(11))/ETH
      PR(18)=(WSA-ONN)/FLOAT(NPC)
      ECK=PR(18)
      ONN=WSA
      NPC=0
      WRITE (6,8116)PROB,NC,T,DTNA,TRAD,DTRAD,NR,N1,N2,N3,N4
      WRITE (6,8117)(PR(I),I=1,8)
      WRITE (6,8118)(PR(I),I=9,12)
      WRITE (6,8119)RADEB,RADER,RADET,UVMAX,ETH,ECK
      II=S8
      IX=S10
      WRITE(6,9040)N10,N11,I1,I2,II,IX
 6090 IF(SENSE LIGHT 4)5000,136
C**** END OF   S P  SUBROUTINE ***********************************************
C
C
5000  CONTINUE
 5001 WRITE (6,8116)PROB,NC,T,DTNA,TRAD,DTRAD,NR,N1,N2,N3,N4
 5004 DO 5050 I=1,I1
      SENSE LIGHT 4
      J=I2+1
      K=I2*IMAX+1+I
      DO 5046 L=1,I2
      J=J-1
      K=K-IMAX
 5012 IF(AMX(K))9917,5046,5014
 5014 IF(SENSE LIGHT 4)5016,5019
 5016 WRITE (6,8135)I,X(I),DX(I)
 5019 WS=AMX(K)/(TAU(I)*DY(J))
      WSA=THETA(K)
 5018 WRITE(6,8108)J,U(K),V(K),P(K),WSA,AMX(K),AIX(K),WS,Y(J)
 5046 CONTINUE
      WRITE (6,9004)
      J = I2+1
      K = I2*IMAX+I+1
      DO 5044 L=1,I2
      J = J-1
      K = K-IMAX
      IF (AMX(K)) 9917,5044,5022
 5022 WRITE(6,9000)J,AIA(K),AMB(K),AMC(K),AMD(K),AIE(K),AMF(K),AMG(K)
 5044 CONTINUE
      IF (CYCLE - CSTOP) 136,5050,136
 5050 CONTINUE
  136 IF (ABSF(ECK)-DMIN)140,140,9905
C**** END OF   L P  SUBROUTINE ***********************************************
```

```
C
C
C                    ERROR
 9901 NK=110
      GO TO 9999
C                    ENERGY CHECK
 9905 NK=136
      GO TO 9999
C                    NEGATIVE MASS
 9917 NK=6015
      GO TO 9999
 9920 NK=904
      GO TO 9999
 9921 NK=912
      GO TO 9999
 9922 NK=918
      GO TO 9999
 9923 NK=922
      GO TO 9999
 9924 NK=926
 9999 NR=6
      CALL ERPER(NR,NK)
10000 RETURN
C
C                    FORMATS
 8100 FORMAT(I4,2E14.6,3E13.6,E14.6,E13.6,E14.6)
 8110 FORMAT(8H1PROBLEM6X,5HCYCLE9X,4HTIME13X,2HDT13X,4HTRAD11X,5HDTRAD1
     12X,2HNR8X,2HN14X,2HN24X,2HN34X,2HN4/(F7.1,I11,2X,1P4E15.7,I10,2X,4
     2I6))
 8117 FORMAT(1HC//17X2HAI16X,2HAK14X,5HAI+AK15X,2HAM/4H DOT3X,1P4E18.7/3
     1H   X4X,4E18.7)
 8118 FORMAT(12X,13H------------5X,13H------------5X,13H------------5
     1X,13H------------/7H TOTALS1P4E18.7)
 8119 FORMAT(2HC //16X*RADEB*13X*RADER*13X*RADET*12X*4AX VEL*13X*ETH*12X
     1*REL ERROR*/7X5E18.7////)
 8120 FORMAT (1HJ///* TAPE4 DUMP ON CYCLE*I5////)
 8124 FORMAT(3H  K12X,5HAM(K)11X,9HSUM AM(K)11X,4HP(K)11X,4HJ(K)/(I3,4X,
     11P4E18.7))
 8131 FORMAT(1H ///11H DY(J) J=1,I2//(10F12.3))
 8132 FORMAT(1H ///11H  Y(J) J=0,I2//(10F12.3))
 8135 FORMAT(1H ///44 I =I3,6X,6HX(I) =F12.3,6X,7HDX(I) =F12.3//3H  JBX,
     1*X*13X*Y*14X*P*13X*THETA*10X*AMX*11X*AIX*12X*RHO*11X*Y*/)
 8201 FORMAT(I10,2H I54A2)
 8202 FORMAT(10X,2H I54A2)
 8211 FORMAT(F7.1,I3,2H I54A2)
 8222 FORMAT(F7.1,3X,2H I54A2)
 8302 FORMAT(I12,10I11)
 8307 FORMAT(5H A1 =1PE12.6,3X,4HA2 =E12.6,3X,6HAMAX =E12.6,6X,4HA1 =E12
     1.6,3X,4HA2 =E12.6,3X,6HAMAX =E12.6)
 8308 FORMAT(1H /)
 8400 FORMAT(I4,7E15.6)
 8404 FORMAT (1H /,2X,*J*,8X,*A*,14X,*B*,14X,*C*,14X,*D*,14X,*E*,14X,
     1 *F*,14X,*G*, /)
 9040 FORMAT(1H / 6I6)
 8999 FORMAT(I5,1X,1P5E12.6)
      END
```

```
      SUBROUTINE TYPE(N)
      CALL TIME (26HMOUNT NEW TAPE IN PLACE OF  )
      CALL DISPLA(5HTAPE ,N)
      PAUSE
      RETURN
      END



      SUBROUTINE POWER(X,XX,N)
      XX=ALOGF(ABSF(X))/2.302[illegible]
      N=XX
      IF(XX)1,2,2
    1 N=N-1
    2 XX=X/10.**N
      RETURN
      END



      SUBROUTINE STATE7
C                D  I  M  E  N  S  I  O  N
C
      DIMENSION AIX(6200),AMX(6200),P(6200),U(6200),V(6200),
     1AMA(6200),AMB(6200),AMC(6200),AMD(6200),AME(6200),AMF(6200),
     2AMG(6200),THETA(6200),DY(200),XX(101),YY(201)
      DIMENSION X(100),    Y(200),    Z(150),    IZ(150),   FLEFT(200)
      DIMENSION TAU(1[illegible]),  PL(200),   UL(200),   PR(200),   UR(200)
      DIMENSION IW1(15),   YAMC(200), SIGC(200), GAMC(200), DX(100)
      COMMON           Z       ,XX       ,UR       ,PR       ,YY
      COMMON           AIR     ,AIX      ,AM       ,THETA    ,AMX      ,AREA
      COMMON           BIG     ,BOUNCE   ,DDXN     ,DDVK     ,DKE      ,DVK
      COMMON           DX      ,DY       ,E        ,FD       ,FS       ,FX
      COMMON           DUT     ,P        ,PABOVE   ,PBLO     ,PIDTS    ,PPABOV
      COMMON           PRM     ,PUL      ,RDT      ,RC       ,REZ      ,RHO
      COMMON           RL,R2,SIG,[illegible]FL,SWITCH ,TABLM
      COMMON           TAU     ,TAUDTS   ,TAUDTX   ,U        ,UK       ,URR
      COMMON           UT      ,UU       ,UUU      ,UTEF     ,UVMAX    ,V
      COMMON           VABOVE  ,VBLO     ,VEL      ,VK       ,VT       ,VTEF
      COMMON           VV      ,VVABOV   ,VVBLO    ,W2       ,W3       ,WPS
      COMMON           WS      ,WSA      ,WSB      ,WSC      ,XL       ,XLF
      COMMON           XN      ,XR       ,YL       ,YLA      ,YN       ,YU
      COMMON           ZMAX    ,I        ,II       ,IN       ,IR       ,IWS
      COMMON           IWSA    ,IWSB     ,IWSC     ,IW1      ,J        ,JN
      COMMON           JP      ,JR       ,K        ,KN       ,KP       ,KR
      COMMON           KR1     ,L        ,M        ,MA       ,MB       ,MC
      COMMON           MD      ,ME       ,MZ       ,N        ,NK       ,NKMAX
      COMMON           NK1     ,NO       ,NR       ,TW2
      COMMON  AMA,AMB,AMC,AMD,AME,AMF,AMG
```

```
C
C
C                    E  Q  U  I  V  A  L  E  N  C  E
C
      0EQUIVALENCE      (Z,IZ,PROB),      (Z(2),CYCLE),     (Z(3),DT),
     1(Z(4),PRINTS),    (Z(5),PRINTL),    (Z(6),DUMPT7),    (Z(7),CSTOP),
     2(Z(8),PIDY),      (Z(9),THZ),       (Z(10),GAM),      (Z(11),GAMD),
     3(Z(12),GAMX),     (Z(13),ETH),      (Z(14),FFA),      (Z(15),FFB),
     4(Z(16),THDZ),     (Z(17),THXZ),     (Z(18),KMAX),     (Z(19),TAMAX),
     5(Z(20),TYMAX),    (Z(21),AMDM),     (Z(22),AMXM),     (Z(23),DNN),
     6(Z(24),DMIN),     (Z(25),FEF),      (Z(26),DTNA),     (Z(27),CVIS),
     7(Z(28),NPR),      (Z(29),NPRI),     (Z(30),NC),       (Z(31),NPC),
     8(Z(32),NRC),      (Z(33),IMAX),     (Z(34),IMAXA),    (Z(35),JMAX),
     9(Z(36),JMAXA),    (Z(37),KMAX),     (Z(38),KMAXA),    (Z(39),NMAX)
      0EQUIVALENCE      (Z(40),ND),       (Z(41),KDT),      (Z(42),IXMAX),
     1(Z(43),NOD),      (Z(44),NOPR),     (Z(45),NIMAX),    (Z(46),NJMAX),
     2(Z(47),I1),       (Z(48),I2),       (Z(49),I3),       (Z(50),I4),
     3(Z(51),N1),       (Z(52),N2),       (Z(53),N3),       (Z(54),N4),
     4(Z(55),N5),       (Z(56),N6),       (Z(57),N7),       (Z(58),N8),
     5(Z(59),N9),       (Z(60),N10),      (Z(61),N11),      (Z(62),NRM),
     6(Z(63),TRAD),     (Z(64),ANRG),     (Z(65),SN),       (Z(66),DXN),
     7(Z(67),RADER),    (Z(68),RADET),    (Z(69),RADEB),    (Z(70),DTRAD),
     8(Z(71),REZFCT),   (Z(72),RSTOP),    (Z(73),SHELL),    (Z(74),BBOUND),
     9(Z(75),TOZONE),   (Z(76),ECK),      (Z(77),SBOUND),   (Z(78),X1)
      0EQUIVALENCE      (Z(79),X2),       (Z(80),Y1),       (Z(81),Y2),
     1(Z(82),CABLN),    (Z(83),VISC),     (Z(84),T),        (Z(85),GMAX),
     2(Z(86),WSGD),     (Z(87),WSGX),     (Z(88),GMADR),    (Z(89),GMAXR),
     3(Z(90),S1),       (Z(91),S2),       (Z(92),S3),       (Z(93),S4),
     4(Z(94),S5),       (Z(95),S6),       (Z(96),S7),       (Z(97),S8),
     5(Z(98),S9),       (Z(99),S10),
     7(Z(130),NFITD),   (Z(131),NFITT),   (Z(132),NFITR),
     8(Z(133),NPARD),   (Z(134),NPART),   (Z(135),NPARR)
C
      EQUIVALENCE(P(2J)),GAMC),(P(400),FAMC)
      EQUIVALENCE       (UR,UL,FLEFT),    (PR,PL,SIGC)
      EQUIVALENCE       (XX(2),X(1)),     (YY(2),Y(1))
C
C
C
C
      M = -2
      LOK = -2
      IF (NC.GE.960.0.AND.NC.LT.962.0) M = NC
      IF (K.EQ. 752) LOK=K
      IF (K.EQ. 753) LOK=K
      IF (K.EQ. 754) LOK=K
      IF (K.EQ. 902) LOK=K
      IF (K.EQ. 903) LOK=K
      IF (K.EQ. 904) LOK=K
      IF (K.EQ. 852) LOK=K
      IF (K.EQ. 853) LOK=K
      IF (K.EQ. 854) LOK=K
      IF (K.EQ. 902) LOK=K
      IF (K.EQ. 903) LOK=K
      IF (K.EQ. 904) LOK=K
      IF (K.EQ. 952) LOK=K
      IF (K.EQ. 953) LOK=K
      IF (K.EQ. 954) LOK=K
```

```
      IF (K.EQ.1002) LOK=K
      IF (K.EQ.1003) LOK=K
      IF (K.EQ.1004) LOK=K
      IF (K.EQ.1052) LOK=K
      IF (K.EQ.1053) LOK=K
      IF (K.EQ.1054) LOK=K
      IF (AIX(K).LT.0.0) GO TO 300
C  COMPUTE VOLUME OF SOLID CARBON IN CELL
      VEL = 0.0
      VOL=TAU(I)*DY
      RHOG=2.3
      VOLG = AMG(K)/RHOG
      IF(NC.EQ.M.AND.K.EQ.LOK)PRINT 8001,VOL,VOLG,NC,K
 8001 FORMAT (1H ,/,2X,*VOL = *,E14.4,2X,*VOLG = *,E14.4,
     1  2X,*NC = *,I4,2X,*K = *,I4)
      IF(VOLG.GT.0.01*VOL) 10,20
   10 VOL=VOL-VOLG
C  COMPUTE VALUES FOR GAMMA AT CELL TEMPERATURE
   20 IF (THETA(K).GT.200.0) 30,28
   28 GMA = 1.4
      GMB = 1.4
      GMC = 1.4
      GMD = 1.33
      GME = 1.33
      GMF = 1.4
      GO TO 31
   30 CALL GAMGAM (THETA(K),GMA,GMB,GMC,GMD,GME,GMF)
C  FIND MAX VALUE OF GAMMA
   31 GMAX = 0.0
      IF(NC.EQ.M.AND.K.EQ.LOK)PRINT 8002,THETA(K),GMA,GMB,GMC,GMD,
     1 GME,GMF
 8002 FORMAT(2X,*THETA(K) = *,E14.4,2X,*GAMMAS ARE *,6(E12.3,2X))
      IF (GMA.GT.GMAX) GMAX=GMA
      IF (GMB.GT.GMAX) GMAX=GMB
      IF (GMC.GT.GMAX) GMAX=GMC
      IF (GMD.GT.GMAX) GMAX=GMD
      IF (GME.GT.GMAX) GMAX=GME
      IF (GMF.GT.GMAX) GMAX=GMF
      WSGX = GMAX-1.0
C  COMPUTE NEW CELL TEMPERATURE
      IF (AMG(K).LT.0.01*AMX(K)) GO TO 86
      IF (THETA(K).GT.1500.) 80,85
   80 CVGB = (4.2+4.09E-4*THETA(K))*4.184E7/12.0
      GO TO  88
   85 CVGB = (0.8+2.555E-3*THETA(K))*4.184E7/12.0
      GO TO 88
   86 CVGB = 0.0
   88 AIG=AIX(K)-AMG(K)*CVGB*(THETA(K)-292.)
      IF(NC.EQ.M.AND.K.EQ.LOK)PRINT 8003,AIG,AIX(K),AMG(K),CVGB,THETA(K)
 8003 FORMAT (2X,*AIG = *,E14.4,2X,*AIX(K) = *,E14.4,2X,*AMG(K) = *,
     1 E14.4,2X,*CVGB = *,E12.3,2X,*THETA(K) = *,E12.3)
      AMOLA=(AMA(K))/(28.0*(GMA-1.0))
      AMOLB=(AMB(K))/(32.0*(GMB-1.0))
      AMOLC=(AMC(K))/(28.0*(GMC-1.0))
      AMOLD=(AMD(K))/(44.0*(GMD-1.0))
      AMOLE=(AME(K))/(18.0*(GME-1.0))
      AMOLF=(AMF(K))/(2.0*(GMF-1.0))
      AMOL=AMOLA+AMOLB+AMOLC+AMOLD+AMOLE+AMOLF
      IF(NC.EQ.M.AND.K.EQ.LOK)PRINT 8004,AMOLA,AMOLB,AMOLC,AMOLD,
```

```
     1 AMOLE,AMOLF,AMOL
 8004 FORMAT (2X,*THE AMOLS ARE *,6(E14.4,2X),*AMOL = *,E14.4)
      IF (AMOL) 89, 93, 90
   89 TEMP = 0.0
      GO TO 92
   90 R = 8.317E7
      TEMP=AIG/(R*AMOL)
   92 CONTINUE
C  BRING GASES AND SOLID CARBON TO TEMPERATURE EQUILIBRIUM
      NTIME = 0
      MTIME = 0
      IF (AMG(K).LT.0.01*AMX(K)) GO TO 200
      IF(ABSF(THETA(K)-TEMP).LE.10.0) GO TO 250
      TEMPG=THETA(K)
      IF(TEMP.LT.TEMPG) GO TO 150
  100 DTEMP = ABSF(TEMP-TEMPG)/6.0
  110 FLAG = 1.0
      TEM=TEMP-0.5*DTEMP
      IF(NC.EQ.M.AND.K.EQ.LOK)PRINT 8000,TEM,TEMP,DTEMP
 8000 FORMAT (2X,*TEM = *,E14.4,2X,*TEMP = *,E14.4,2X,*DTEMP = *,E14.4)
      IF(NC.EQ.M.AND.K.EQ.LOK)PRINT 8010,TEMPG,DTEMPG,K,NTIME,FLAG
 8010 FORMAT(2X,*TEMPG = *,E14.4,2X,*DTEMPG = *,E14.4,2X,2I6,F6.2)
      CALL CVGAM (TEM,CVA,CVB,CVC,CVD,CVE,CVF)
      DQA=AMA(K)*CVA*DTEMP
      DQB=AMB(K)*CVB*DTEMP
      DQC=AMC(K)*CVC*DTEMP
      DQD=AMD(K)*CVD*DTEMP
      DQE=AME(K)*CVE*DTEMP
      DQF=AMF(K)*CVF*DTEMP
      Q=DQA+DQB+DQC+DQD+DQE+DQF
      IF(NC.EQ.M.AND.K.EQ.LOK)PRINT 8015,DQA,DQB,DQC,DQD,DQE,DQF,Q
 8015 FORMAT(2X,6(E14.4,2X),*Q = *,E14.4)
      IF (AMG(K).LE.0.0.OR.CVGB.LE.0.0) GO TO 200
      DTEMPG=Q/(AMG(K)*CVGB)
      IF (DTEMPG.GT.ABSF(TEMP-TEMPG)) 130,140
  130 RAT = (0.9*ABSF(TEMP-TEMPG))/DTEMPG
      DTEMP = DTEMP*RAT
      GO TO 110
  140 TEMP = TEMP-DTEMP
      TEMPG = TEMPG+DTEMPG
      IF(NC.EQ.M.AND.K.EQ.LOK)PRINT 8006,AMG(K),CVGB,DTEMPG,TEMP,TEMPG
 8006 FORMAT (2X,*AMG(K) = *,E14.4,2X,*CVGB = *,E14.4,2X,*DTEMPG = *,
     1 E14.4,2X,*TEMP = *,E14.4,2X,*TEMPG = *,E14.4)
      NTIME = NTIME + 1
      IF (ABSF(TEMP-TEMPG).LE.10.0) GO TO 200
      IF (TEMP.LT.TEMPG) 150,100
  150 DTEMP = ABSF(TEMPG-TEMP)/6.0
  155 FLAG = 2.0
      TEM=TEMP+0.5*DTEMP
      IF(NC.EQ.M.AND.K.EQ.LOK)PRINT 8000,TEM,TEMP,DTEMP,
     1 TEMPG,DTEMPG,K,NTIME,FLAG
      CALL CVGAM (TEM,CVA,CVB,CVC,CVD,CVE,CVF)
      DQA=AMA(K)*CVA*DTEMP
      DQB=AMB(K)*CVB*DTEMP
      DQC=AMC(K)*CVC*DTEMP
      DQD=AMD(K)*CVD*DTEMP
      DQE=AME(K)*CVE*DTEMP
      DQF=AMF(K)*CVF*DTEMP
```

```
      Q=DQA+DQB+DQC+DQD+DQE+DQF
      IF(NC.EQ.M.AND.K.EQ.LOK)PRINT 8005,DQA,DQB,DQC,DQD,DQE,DQF,Q
      IF (AMG(K).LE.0.J.OR.CVGB.LE.0.0) GO TO 200
      DTEMPG=Q/(AMG(K)*CVGB)
      IF (DTEMPG.GT.ABSF(TEMP-TEMPG)) 160,170
  160 RAT = (0.9*ABSF(TEMP-TEMPG))/DTEMPG
      DTEMP = DTEMP*RAT
      GO TO 155
  170 TEMP = TEMP + DTEMP
      TEMPG = TEMPG - DTEMPG
      IF(NC.EQ.M.AND.K.EQ.LOK)PRINT 8006,AMG(K),CVGB,DTEMPG,TEMP,TEMPG
      MTIME = MTIME + 1
      IF (ABSF(TEMP-TEMPG).LE.10.0) GO TO 200
      IF (TEMP.LT.TEMPG) 150,100
  200 IF (AMG(K).LT.0.01*AMX(K)) GO TO 205
      TEMP = (TEMP + TEMPG)*0.5
      TEMPG = TEMP
  205 IF (ABSF(THETA(K)-TEMP).LE.10.0) GO TO 250
  210 THETA(K) = THETA(K) + 0.9*(TEMP - THETA(K))
      VEL = VEL + 1.0
      IF(NC.EQ.M.AND.K.EQ.LOK)PRINT 8007,THETA(K),TEMP,VEL
 8007 FORMAT (2X,*THETA(K) = *,E14.4,2X,*TEMP = *,E14.4,2X,*VEL = *,
     1 F6.2)
      GO TO 20
  300 P(K) = 0.0
      THETA(K) = 50.0
      PRINT 8009,K
 8009 FORMAT (2X,*NEGATIVE AIX IN CELL*,I6,*PRESSURE SET TO ZERO*)
      GO TO 10000
  250 THETA(K) = 0.5*(THETA(K) + TEMP)
      RTV = (R*THETA(K))/VOL
      PA = (AMA(K)*RTV)/28.0
      PB = (AMB(K)*RTV)/32.0
      PC = (AMC(K)*RTV)/28.0
      PD = (AMD(K)*RTV)/44.0
      PE = (AME(K)*RTV)/18.0
      PF = (AMF(K)*RTV)/2.0
      P(K) = PA+PB+PC+PD+PE+PF
      IF(NC.EQ.M.AND.K.EQ.LOK)PRINT 8008,THETA(K),P(K),PA,PB,PC,PD,PE,PF
 8008 FORMAT (1X,*THETA = *,E14.4,1X,*P = *,E14.4,1X,*PRESS. ARE *,
     1 6(E12.3,1X))
10000 RETURN
      END
```

```
      SUBROUTINE CVGAM (TMP,CA,CB,CC,CD,CE,CF)
      T = TMP
      CALERG=4.184E7
      IF (T.LT.50.0) T = 50.0
      IF (T.GT.6000.) T = 6000.0
      TT=T*T
      TTT = T*TT
      CA=(4.4163+1.7675E-3*T-4.0340E-7*TT+3.0933E-11*TTT)*CALERG/28.0
      CB=(4.6080+1.9774E-3*T-3.6968E-7*TT+2.6853E-11*TTT)*CALERG/32.0
      CC=(4.4239+1.8638E-3*T-4.4233E-7*TT+3.4856E-11*TTT)*CALERG/28.0
      CD=(5.8423+5.9368E-3*T-1.5217E-6*TT+1.2629E-10*TTT)*CALERG/44.0
      CE=(4.7832+3.9438E-3*T-7.9342E-7*TT+5.4627E-11*TTT)*CALERG/18.0
      CF=(4.5507+9.0598E-4*T-4.9424E-8*TT-3.1041E-12*TTT)*CALERG/2.0
  100 RETURN
      END




      SUBROUTINE GAMGAM (TMP,GA,GB,GC,GD,GE,GF)
      T = TMP
      IF (T.LT.50.0) T = 50.0
      IF (T.GT.6000.0) T = 6000.0
      TT=T*T
      TTT = T*TT
      CALERG = 4.184E7
      CPA=(6.3930+1.7789E-3*T-4.1020E-7*TT+3.1875E-11*TTT)*CALERG/28.0
      CPB=(6.6246+1.8603E-3*T-3.6215E-7*TT+2.5902E-11*TTT)*CALERG/32.0
      CPC=(6.4114+1.8637E-3*T-4.4326E-7*TT+3.5074E-11*TTT)*CALERG/28.0
      CPD=(7.8268+5.9673E-3*T-1.5678E-6*TT+1.2213E-10*TTT)*CALERG/44.0
      CPE=(6.7246+3.9967E-3*T-7.4975E-7*TT+4.9546E-11*TTT)*CALERG/18.0
      CPF=(6.5543+8.6152E-4*T-2.7312E-8*TT-3.1196E-12*TTT)*CALERG/2.0
      CALL CVGAM (T,CVA,CVB,CVC,CVD,CVE,CVF)
      GA = CPA/CVA
      GB = CPB/CVB
      GC = CPC/CVC
      GD = CPD/CVD
      GE = CPE/CVE
      GF = CPF/CVF
      RETURN
      END
```

```
      SUBROUTINE INTFACE
C
CC                D I M E N S I O N
C
      DIMENSION AIX(6200),AMX(6200),P(6200),U(6200),V(6200),
     1AMA(6200),AMB(6200),AMC(6200),AMD(6200),AME(6200),AMF(6200),
     2AMG(6200),THETA(6200),DF(200),XX(101),FF(201)
      DIMENSION X(100),    Y(200),    Z(150),    IZ(150),   FLEFT(200)
      DIMENSION TAU(100),  PL(200),   UL(200),   PR(200),   UR(200)
      DIMENSION IW1(15),   VAMC(200), SIGC(200), GAMC(200), DX(100)
C
      COMMON          Z      ,XX     ,UR     ,PR     ,YY
      COMMON          AID    ,AIX    ,AM     ,THETA  ,AMX    ,AREA
      COMMON          BIG    ,BOUNCE ,DDXN   ,DDVK   ,DKE    ,DVK
      COMMON          DX     ,DF     ,E      ,FD     ,FS     ,FX
      COMMON          OUT    ,P      ,PABOVE ,PBLO   ,PIOTS  ,PPABOV
      COMMON          PRR    ,PUL    ,RDT    ,RC     ,REZ    ,RHO
      COMMON          RL,RR,SIG,QQQQFL,SWITCH ,TABLM
      COMMON          TAU    ,TAUDTS ,TAUDTX ,U      ,UK     ,URR
      COMMON          UT     ,UU     ,UUU    ,UTEF   ,UVMAX  ,V
      COMMON          VABOVE ,VB_O   ,VE_    ,VK     ,VT     ,VTEF
      COMMON          VV     ,VVABOV ,VVBLO  ,W2     ,W3     ,WPS
      COMMON          WS     ,WSA    ,WSB    ,WSC    ,XL     ,XLF
      COMMON          XN     ,XR     ,YL     ,Y_W    ,YN     ,YU
      COMMON          ZMAX   ,I      ,II     ,IN     ,IR     ,IWS
      COMMON          IWSA   ,IWSB   ,IWSC   ,IW1    ,J      ,JN
      COMMON          JP     ,JR     ,K      ,KN     ,KP     ,KR
      COMMON          KRM    ,L      ,M      ,MA     ,MB     ,MC
      COMMON          MD     ,ME     ,MZ     ,N      ,NK     ,NKMAX
      COMMON          NK1    ,NO     ,NR     ,IW2
      COMMON  AMA,AMB,AMC,AMD,AME,AMF,AMG
C
CC
CC                E Q U I V A L E N C E
C
      EQUIVALENCE     (Z,IZ,PROB),     (Z(2),CYCLE),    (Z(3),DT),
     1(Z(4),PRINTS),  (Z(5),PRINTL),   (Z(6),DUMPT),    (Z(7),CSTOP),
     2(Z(8),PIOV),    (Z(9),TMZ),      (Z(10),GAM),     (Z(11),GAMD),
     3(Z(12),GAMX),   (Z(13),ETH),     (Z(14),FFA),     (Z(15),FFB),
     4(Z(16),TMDZ),   (Z(17),TMXZ),    (Z(18),XMAX),    (Z(19),TXMAX),
     5(Z(20),TYMAX),  (Z(21),AMDM),    (Z(22),AMXM),    (Z(23),DNN),
     6(Z(24),DMIN),   (Z(25),FEF),     (Z(26),DTNA),    (Z(27),CVIS),
     7(Z(28),NPR),    (Z(29),NPRI),    (Z(30),NC),      (Z(31),NPC),
     8(Z(32),NRC),    (Z(33),IMAX),    (Z(34),IMAXA),   (Z(35),JMAX),
     9(Z(36),JMAXA),  (Z(37),KMAX),    (Z(38),KMAXA),   (Z(39),NMAX)
      EQUIVALENCE     (Z(40),ND),      (Z(41),KDT),     (Z(42),IXMAX),
     1(Z(43),NOD),    (Z(44),NOPR),    (Z(45),NIMAX),   (Z(46),NJMAX),
     2(Z(47),I1),     (Z(48),I2),      (Z(49),I3),      (Z(50),I4),
     3(Z(51),N1),     (Z(52),N2),      (Z(53),N3),      (Z(54),N4),
     4(Z(55),N5),     (Z(56),N6),      (Z(57),N7),      (Z(58),N8),
     5(Z(59),N9),     (Z(60),N10),     (Z(61),N11),     (Z(62),NRM),
     6(Z(63),TRAD),   (Z(64),XNRG),    (Z(65),SN),      (Z(66),DXN),
     7(Z(67),RADERD), (Z(68),RADETD),  (Z(69),RADESD),  (Z(70),DTRADD),
     8(Z(71),REZFCT), (Z(72),RSTOP),   (Z(73),SMEL_),   (Z(74),BBOJND),
     9(Z(75),TOZONE), (Z(76),ECK),     (Z(77),SBOJN),   (Z(78),X1)
      EQUIVALENCE     (Z(79),X2),      (Z(80),Y1),      (Z(81),Y2),
     1(Z(82),CABLN),  (Z(83),VISC),    (Z(84),T),       (Z(85),GMAX),
     2(Z(86),WSGD),   (Z(87),WSGX),    (Z(88),GMAXR),   (Z(89),GMAXR),
```

```
     3(Z(90),S1),      (Z(91),S2),      (Z(92),S3),      (Z(93),S4),
     4(Z(94),S5),      (Z(95),S6),      (Z(96),S7),      (Z(97),S8),
     5(Z(98),S9),      (Z(99),S10),
     7(Z(130),NFITB),  (Z(131),NFITT),  (Z(132),NFITR),
     8(Z(133),NPARB),  (Z(134),NPART),  (Z(135),NPARR)
C
      EQUIVALENCE(P(200),GAMC),(P(400),YAMC)
      EQUIVALENCE      (UR,UL,FLEFT),    (PR,PL,SIGC)
      EQUIVALENCE      (XX(2),X(1)),     (YY(2),Y(1))
      EQUIVALENCE (Z(105),RATA), (Z(106),RATB), (Z(107),RATC)
      EQUIVALENCE (Z(109),RATD), (Z(109),RATE), (Z(110),RATF)
      EQUIVALENCE (Z(111),RATG), (Z(112),ENIA), (Z(113),ENIB)
      EQUIVALENCE (Z(114),ENIC), (Z(115),ENID), (Z(116),ENIE)
      EQUIVALENCE (Z(117),ENIF), (Z(118),ENIG)
      EQUIVALENCE (Z(120),DETVEL), (Z(121),ENSUM), (Z(122),EXEN)
      EQUIVALENCE (Z(123),EXPHO), (Z(124),THICK), (Z(125),AJCEL)
      EQUIVALENCE (Z(126),TOTLEN)
C
C
      KNIT = 14
      NIT = 0
      JCELL = AJCEL
      JCELL = JCELL - 4
      KCELL = 2 + (JCELL - 1)*IMAX
   10 KKCEL = KCELL + 1
      IF (AMC(KKCEL).GT.0.0001*AMX(KKCEL)) GO TO 20
      IF (AMD(KKCEL).GT.0.0001*AMX(KKCEL)) GO TO 20
      IF (AME(KKCEL).GT.0.0001*AMX(KKCEL)) GO TO 20
      IF (AMF(KKCEL).GT.0.0001*AMX(KKCEL)) GO TO 20
      IF (AMG(KKCEL).LE.0.0001*AMX(KKCEL)) GO TO 100
C ****** THE QUANTITY OF AIR IN A CELL IS TAKEN TO BE 4.083 TIMES
C ****** THE MASS OF OXYGEN IN THE CELL
   20 AIRMAS = AMB(KCELL)*4.083
      PRODMAS = AMX(KKCEL) - 4.083*AMB(KKCEL)
      IF (AIRMAS.LE.PRODMAS)  GO TO 30
      BBTRAN = PRODMAS/4.083
      AATRAN = 3.083*BBTRAN
      CTRAN = AMC(KKCEL)
      DTRAN = AMD(KKCEL)
      ETRAN = AME(KKCEL)
      FTRAN = AMF(KKCEL)
      GTRAN = AMG(KKCEL)
      ATRAN = AMA(KKCEL) - 3.083*AMB(KKCEL)
      GO TO 40
   30 BBTRAN = AMB(KCELL)
      AATRAN = 3.083*BBTRAN
      FACTR = AIRMAS/PRODMAS
      CTRAN = FACTR*AMC(KKCEL)
      DTRAN = FACTR*AMD(KKCEL)
      ETRAN = FACTR*AME(KKCEL)
      FTRAN = FACTR*AMF(KKCEL)
      GTRAN = FACTR*AMG(KKCEL)
      ATRAN = FACTR*(AMA(KKCEL) - 3.083*AMB(KKCEL))
   40 AMA(KKCEL) = AMA(KKCEL) + AATRAN - ATRAN
      AMB(KKCEL) = AMB(KKCEL) + BBTRAN
      AMA(KCELL) = AMA(KCELL) - AATRAN + ATRAN
      AMB(KCELL) = AMB(KCELL) - BBTRAN
      AMC(KCELL) = AMC(KCELL) + CTRAN
```

```
      AMC(KKCEL) = AMC(KKCEL) - CTRAN
      AMD(KCELL) = AMD(KCELL) + DTRAN
      AMD(KKCEL) = AMD(KKCEL) - DTRAN
      AME(KCELL) = AME(KCELL) + ETRAN
      AME(KKCEL) = AME(KKCEL) - ETRAN
      AMF(KCELL) = AMF(KCELL) + FTRAN
      AMF(KKCEL) = AMF(KKCEL) - FTRAN
      AMG(KCELL) = AMG(KCELL) + GTRAN
      AMG(KKCEL) = AMG(KKCEL) - GTRAN
      TMASK = AMA(KCELL)+AMB(KCELL)+AMC(KCELL)+AMD(KCELL)+AME(KCELL)
     1    +AMF(KCELL)+AMG(KCELL)
      TMASKK = AMA(KKCEL)+AMB(KKCEL)+AMC(KKCEL)+AMD(KKCEL)+AME(KKCEL)
     1   +AMF(KKCEL)+AMG(KKCEL)
      IF (ABS(TMASK-AMX(KCELL)).GT.1.0E-6)  GO TO 200
      IF (ABS(TMASKK-AMX(KKCEL)).GT.1.0E-6)  GO TO 200
      KCELL = KCELL + 1
      GO TO 10
  100 NIT = NIT + 1
      IF (NIT.GE.KNIT) GO TO 150
      KCELL = KCELL + IMAX
      GO TO 10
  150 PRINT 8800
 8800 FORMAT ( 10X,* SUBROUTINE INTERFACE COMPLETED *)
      GO TO 300
  200 PRINT 8801
 8801 FORMAT (10X,* MASS BALLANCE NOT CORRECT *)
  300 RETURN
      END
```

```
      SUBROUTINE POC
C
CC                D I M E N S I O N
C
      DIMENSION AIX(6200),AMX(6200),P(6200),U(6200),V(6200),
     1AMA(6200),AMB(6200),AMC(6200),AMD(6200),AME(6200),AMF(6200),
     2AMG(6200),THETA(6200),DY(200),XX(101),YY(201)
      DIMENSION X(100),     F(200),     Z(150),     IZ(150),    FLEFT(200)
      DIMENSION TAU(100),   PL(200),    UL(200),    PR(200),    UR(200)
      DIMENSION IW1(15),    YAMC(200),  SIGC(200),  GAMC(200),  DX(100)
      COMMON            Z       ,XX     ,UR     ,PR     ,YY
      COMMON            AID     ,AIX    ,AM     ,THETA  ,AMX    ,AREA
      COMMON            BIG     ,BOUNCE ,DDKN   ,DDVK   ,DKE    ,DVK
      COMMON            DX      ,DY     ,E      ,FD     ,FS     ,FX
      COMMON            OUT     ,P      ,PABOVE ,PBLO   ,PIDTS  ,PPABOV
      COMMON            PRR     ,PUL    ,QDT    ,RC     ,REZ    ,RHO
      COMMON            RL,RR,SIG,QQQDFL,SWITCH ,TABLM
      COMMON            TAU     ,TAUDTS ,TAUDTX ,U      ,UK     ,URR
      COMMON            UT      ,UU     ,UUU    ,UTEF   ,UVMAX  ,V
      COMMON            VABOVE  ,VBLO   ,VEL    ,VK     ,VT     ,VTEF
      COMMON            VV      ,VVABOV ,VVBLO  ,W2     ,W3     ,WPS
      COMMON            WS      ,WSA    ,WSB    ,WSC    ,XL     ,XLF
      COMMON            XN      ,XP     ,YL     ,YLW    ,YN     ,YU
      COMMON            ZMAX    ,I      ,II     ,IN     ,IR     ,IWS
      COMMON            IWSA    ,IWSB   ,IWSC   ,IW1    ,J      ,JN
      COMMON            JP      ,JR     ,K      ,KN     ,KP     ,KR
      COMMON            KRM     ,L      ,M      ,MA     ,MB     ,MC
      COMMON            MD      ,ME     ,MZ     ,N      ,NK     ,NKMAX
      COMMON            NK1     ,NO     ,NR     ,IW2
      COMMON   AMA,AMB,AMC,AMD,AME,AMF,AMG
C
CC
CC                E Q U I V A L E N C E
C
      EQUIVALENCE       (Z,IZ,PROB),      (Z(2),CYCLE),     (Z(3),DT),
     1(Z(4),PRINTS),    (Z(5),PRINTL),    (Z(6),DUMPT7),    (Z(7),CSTOP),
     2(Z(8),PID(),      (Z(9),TMZ),       (Z(10),GAM),      (Z(11),GAMD),
     3(Z(12),GAMX),     (Z(13),ETH),      (Z(14),FFA),      (Z(15),FFB),
     4(Z(16),TMDZ),     (Z(17),TMZ),      (Z(18),XMAX),     (Z(19),TXMAX),
     5(Z(20),TYMAX),    (Z(21),AMDM),     (Z(22),AMX4),     (Z(23),DNN),
     6(Z(24),DMIN),     (Z(25),FEF),      (Z(26),DTNA),     (Z(27),CVIS),
     7(Z(28),NPR),      (Z(29),NPRI),     (Z(30),NC),       (Z(31),NPC),
     8(Z(32),NPC),      (Z(33),IMAX),     (Z(34),IMAXA),    (Z(35),JMAX),
     9(Z(36),JMAXA),    (Z(37),KMAX),     (Z(38),KMAXA),    (Z(39),NMAX)
      EQUIVALENCE       (Z(40),ND),       (Z(41),KDT),      (Z(42),IXMAX),
     1(Z(43),NOD),      (Z(44),NOPR),     (Z(45),NIMAX),    (Z(46),NJMAX),
     2(Z(47),I1),       (Z(48),I2),       (Z(49),I3),       (Z(50),I4),
     3(Z(51),N1),       (Z(52),N2),       (Z(53),N3),       (Z(54),N4),
     4(Z(55),N5),       (Z(56),N6),       (Z(57),N7),       (Z(58),N8),
     5(Z(59),N9),       (Z(60),N10),      (Z(61),N11),      (Z(62),NRM),
     6(Z(63),TRAD),     (Z(64),XNRG),     (Z(65),SN),       (Z(66),DKN),
     7(Z(67),RADER),    (Z(68),RADET),    (Z(69),RADEB),    (Z(70),DTRAD),
     8(Z(71),REZFCT),   (Z(72),RSTOP),    (Z(73),SHELL),    (Z(74),BBOUND),
     9(Z(75),TOZONE),   (Z(76),ECK),      (Z(77),SBOUND),   (Z(78),X1)
      EQUIVALENCE       (Z(79),X2),       (Z(80),Y1),       (Z(81),Y2),
     1(Z(82),CABLN),    (Z(83),VISC),     (Z(84),T),        (Z(85),GMAX),
     2(Z(86),WSGD),     (Z(87),WSGX),     (Z(88),GMAXR),    (Z(89),GMAXR),
     3(Z(90),S1),       (Z(91),S2),       (Z(92),S3),       (Z(93),S4),
```

```
     4(Z(94),S5),       (Z(95),S6),       (Z(96),S7),       (Z(97),S8),
     5(Z(98),S9),       (Z(99),S10),
     7(Z(130),NFIT1),   (Z(131),NFITT),   (Z(132),NFIT2),
     8(Z(133),NPAR3),   (Z(134),NPART),   (Z(135),NPAR2)
C
      EQUIVALENCE(P(200),GAMC),(P(400),YAMC)
      EQUIVALENCE       (UR,U_,FLEFT),    (PR,PL,SIGC)
      EQUIVALENCE       (XX(2),X(1)),     (YY(2),Y(1))
      EQUIVALENCE (Z(100),PDCSUM)
C
C
C
C
      LOK = -2
      M = -2
      IF (NC.GE.700.).AND.NC.LE.750.0) M = NC
      PDCSUM = 0.0
      VOLM = 4.5
      DO 1200 K=2,K14X
      NIT = 1
      TEMPEN = 0.0
      TG = TM*TA(K)
      TB = TG
      IF (TG.GT.200).0) TB=3[illegible]0.+0.95*TG-5.0E-05*TG*TG
      KOXC = 0
      KOXCO = 0
      KOXH = 0
      IF (K.EQ. 752) LOK=K
      IF (K.EQ. 753) LOK=K
      IF (K.EQ. 754) LOK=K
      IF (K.EQ. 802) LOK=K
      IF (K.EQ. 803) LOK=K
      IF (K.EQ. 804) LOK=K
      IF (K.EQ. 852) LOK=K
      IF (K.EQ. 853) LOK=K
      IF (K.EQ. 854) LOK=K
      IF (K.EQ. 902) LOK=K
      IF (K.EQ. 903) LOK=K
      IF (K.EQ. 904) LOK=K
      IF (K.EQ. 952) LOK=K
      IF (K.EQ. 953) LOK=K
      IF (K.EQ. 954) LOK=K
      IF (K.EQ.1002) LOK=K
      IF (K.EQ.1003) LOK=K
      IF (K.EQ.1004) LOK=K
      IF (K.EQ.1052) LOK=K
      IF (K.EQ.1053) LOK=K
      IF (K.EQ.1054) LOK=K
      IF (K.EQ.2500) LOK = K
      IF (K.EQ.3500) LOK = K
      IF (K.EQ.4600) LOK = K
      IF (NC.EQ.1.AND.K.EQ.LOK) PRINT 3000,NIT,AMX(K),AMA(K),AMB(K),
     1  AMC(K),AMD(K),AME(K),AMF(K),AMG(K),TEMPEN,AIX(K)
      IF (AMX(K).LE.0.0) GO TO 1200
C ***** TEST FOR CAPABILITY OF COMBUSTION IN CELL
      IF (AMB(K).LT.0.001*AMX(K)) GO TO 1000
      IF (AMG(K).GE.0.001*AMX(K)) GO TO 5
      IF (AMF(K).GE.0.001*AMX(K)) GO TO 5
```

```
      IF (AMC(K).LT.0.001*AMX(K)) GO TO 1000
C ***** COMPUTE REACTION RATES AT CELL TEMPERATURE
    5 RAT12 = 1.9E+16/SQRT(TG)
      RAT11 = RAT12*EXP(-5.92E04/TB)
      RAT02 = 1.0E+16*EXP(-3.5E+03/(8.31E+07*TG))
      RAT04 = 1.0E+19/TG
      RAT06 = 3.0E+16/SQRT(TG)
C ***** COMPUTE AMOUNT OF ATOMIC OXYGEN AVAILABLE FOR REACTION
      AMOLE = AMA(K)/(28.0*VOLM)
      BMOLE = AMB(K)/(32.0*VOLM)
      CMOLE = AMC(K)/(28.0*VOLM)
      DMOLE = AMD(K)/(44.0*VOLM)
      EMOLE = AME(K)/(18.0*VOLM)
      FMOLE = AMF(K)/(2.0*VOLM)
      GMOLE = AMG(K)/(12.0*VOLM)
      TMOLE = AMOLE+BMOLE+CMOLE+DMOLE+EMOLE+FMOLE+GMOLE
      DOX = 2.0*RAT11*BMOLE*TMOLE*DT
      OX = VOLM*DOX*16.0
      NIT = 6
      IF (NC.EQ.1.AND.K.EQ.LOK) PRINT 8003,K,THETA(K),TG,RAT11,
     1  RAT12,TMOLE,BMOLE,DOX,OX
      IF (NC.EQ.1.AND.K.EQ.LOK) PRINT 8000,NIT,AMX(K),AMA(K),AMB(K),
     1  AMC(K),AMD(K),AME(K),AMF(K),AMG(K),TEMPEN,OX
      IF (OX.GT.AMB(K))  OX = AMB(K)
C ***** DETERMINE WHICH FUELS ARE IN CELL
      IF (AMC(K)+AMF(K).LE.0.002*AMX(K)) GO TO 10
      IF (AMC(K)+AMG(K).LE.0.002*AMX(K)) GO TO 20
      IF (AMF(K)+AMG(K).LE.0.002*AMX(K)) GO TO 30
C ***** COMPUTE QUANTITY OF ATOMIC HYDROGEN
      HOX = SQRT(FMOLE*EXP(-5.22E-04/TG))
      OH = 1.5E-08
C ***** COMPUTE RATIOS OF PRODUCTION OF H2O,CO2,AND CO
      H2O = RAT04*OH*HOX*TMOLE
      CO = RAT06*GMOLE*DOX*TMOLE
      COO = RAT02*CMOLE*DOX*TMOLE
C ***** TEST TO DETERMINE IF ALL THREE FUELS ARE PRESENT IN THE CELL
      IF (AMG(K).LE.0.001*AMX(K)) GO TO 40
      IF (AMF(K).LE.0.001*AMX(K)) GO TO 40
      IF (AMC(K).LE.0.001*AMX(K)) GO TO 40
C ***** COMPUTE THE AMOUNT OF ATOMIC OXYGEN GOING TO EACH REACTION
      RSUM = H2O + CO + COO
      OXC = (COO*OX)/RSUM
      OXCO = (COO*OX)/RSUM
      OXH = (H2O*OX)/RSUM
C ***** TEST FOR ENOUGH OF EACH FUEL TO BURN THE ABOVE OXYGEN
    7 AMGG = OXC/1.334
      IF (AMG(K).LE.1.0E-15) GO TO 9
      IF (AMGG.GT.AMG(K)) GO TO 80
    9 AMFF = OXH/8.0
      IF (AMF(K).LE.1.0E-15) GO TO 9
      IF (AMFF.GT.AMF(K)) GO TO 90
    9 AMCC = OXCO/0.572
      IF (AMC(K).LE.1.0E-15) GO TO 11
      IF (AMCC.GT.AMC(K)) GO TO 100
   11 NIT = 7
      IF (NC.EQ.1.AND.K.EQ.LOK) PRINT 8000,NIT,AMX(K),AMA(K),AMB(K),
     1  AMC(K),AMD(K),AME(K),AMF(K),AMG(K),TEMPEN,OX
C ***** ACCOMPLISH ACTUAL BURN REACTIONS
```

```
      TEMPEN = TEMPEN + 9.3E10*AMGG
      AMG(K) = AMG(K) - AMGG
      AMB(K) = AMB(K) - OXC
      AMC(K) = AMX(K)-AMA(K)-AMB(K)-AMD(K)-AME(K)-AMF(K) - AMG(K)
      TEMPEN = TEMPEN + 1.42E12*AMFF
      AMF(K) = AMF(K) - AMFF
      AMB(K) = AMB(K) - OXM
      AME(K) = AMX(K)-AMA(K)-AMB(K)-AMC(K)-AMD(K)-AMF(K)-AMG(K)
      TEMPEN = TEMPEN + 1.02E11*AMCC
      AMC(K) = AMC(K) - AMCC
      AMB(K) = AMB(K) - OXCO
      AMD(K) = AMX(K)-AMA(K)-AMB(K)-AME(K)-AMC(K)-AMF(K)-AMG(K)
      NIT = 2
      IF (NC.EQ.M.AND.K.EQ.LOK) PRINT 9000,NIT,AMX(K),AMA(K),AMB(K),
     1  AMC(K),AMD(K),AME(K),AMF(K),AMG(K),TEMPEN,OX
      GO TO 1100
   11 IF (AMG(K).LT.OX/1.334) GO TO 12
      AMGG = OX/1.334
      OXMAS = OX
      GO TO 14
   12 AMGG = AMG(K)
      OXMAS = 1.334*AMGG
   14 TEMPEN = TEMPEN + 9.3E+10*AMGG
      AMG(K) = AMG(K) - AMGG
      AMB(K) = AMB(K) - OXMAS
      AMC(K) = AMX(K)-AMA(K)-AMD(K)-AME(K)-AMF(K)-AMG(K)-AMB(K)
      NIT = 3
      IF (NC.EQ.1.AND.K.EQ.LOK) PRINT 9000,NIT,AMX(K),AMA(K),AMB(K),
     1  AMC(K),AMD(K),AME(K),AMF(K),AMG(K),TEMPEN,OX
      GO TO 1100
   21 IF (AMF(K).LT.OX/1.0) GO TO 22
      AMFF = OX/1.0
      OXMAS = OX
      GO TO 24
   22 AMFF = AMF(K)
      OXMAS = 1.0*AMFF
   24 TEMPEN = TEMPEN + 1.42E12*AMFF
      AMF(K) = AMF(K)-AMFF
      AMB(K) = AMB(K) - OXMAS
      AME(K) = AMX(K)-AMA(K)-AMC(K)-AMD(K)-AMF(K)-AMB(K)-AMG(K)
      NIT = 4
      IF (NC.EQ.1.AND.K.EQ.LOK) PRINT 9000,NIT,AMX(K),AMA(K),AMB(K),
     1  AMC(K),AMD(K),AME(K),AMF(K),AMG(K),TEMPEN,OX
      GO TO 1100
   31 IF (AMC(K).LT.OX/3.572) GO TO 32
      AMCC = OX/3.572
      OXMAS = OX
      GO TO 34
   32 AMCC = AMC(K)
      OXMAS = 3.572*AMCC
   34 TEMPEN = TEMPEN + 1.02E11*AMCC
      AMC(K) = AMC(K) - AMCC
      AMB(K) = AMB(K) - OXMAS
      AMD(K) = AMX(K)-AMA(K)-AMC(K)-AME(K)-AMB(K)-AMF(K)-AMG(K)
      NIT = 5
      IF (NC.EQ.M.AND.K.EQ.LOK) PRINT 9000,NIT,AMX(K),AMA(K),AMB(K),
     1  AMC(K),AMD(K),AME(K),AMF(K),AMG(K),TEMPEN,OX
      GO TO 1100
```

```
   4  IF (AMG(K).LE.0.001*AMX(K)) GO TO 50
      IF (AMF(K).LE.0.001*AMX(K)) GO TO 60
      GO TO 70
   5) RSUM = H2O + COO
      OXC = 0.0
      KOXC = 1
      OXCO = (COO*OX)/RSUM
      IF (KOXCO.EQ.1) OXCO = 0.0
      OXH = (H2O*OX)/RSUM
      IF (KOXH.EQ.1)  OXH = 0.0
      GO TO 7
   6) RSUM = CO + COO
      OXC = (CO*OX)/RSUM
      IF (KOXC.EQ.1)  OXC = 0.0
      OXCO = (COO*OX)/RSUM
      IF (KOXCO.EQ.1) OXCO = 0.0
      OXH = 0.0
      KOXH = 1
      GO TO 7
   7) RSUM = CO + H2O
      OXC = (CO*OX)/RSUM
      IF (KOXC.EQ.1)  OXC = 0.0
      OXCO = 0.0
      KOXCO = 1
      OXH = (H2O*OX)/RSUM
      IF (KOXH.EQ.1)  OXH = 0.0
      GO TO 7
   8) AMGG = AMG(K)
      OXMAS = 1.334*AMGG
      TEMPEN = TEMPEN + 9.3E+10*AMGG
      AMG(K) = 0.0
      AMB(K) = AMB(K) - OXMAS
      AMC(K) = AMX(K)-AMA(K)-AMB(K)-AMD(K)-AME(K)-AMF(K)-AMG(K)
      OX = OX - OXMAS
      GO TO 30
   9) AMFF = AMF(K)
      OXMAS = 0.5*AMFF
      TEMPEN = TEMPEN + 1.425E+12*AMFF
      AMF(K) = 0.0
      AMB(K) = AMB(K) - OXMAS
      AME(K) = AMX(K)-AMA(K)-AMB(K)-AMC(K)-AMD(K)-AMF(K)-AMG(K)
      OX = OX - OXMAS
      GO TO 60
  10) AMCC = AMC(K)
      OXMAS = 0.572*AMCC
      TEMPEN = TEMPEN + 1.02E+11*AMCC
      AMC(K) = 0.0
      AMB(K) = AMB(K) - OXMAS
      AMD(K) = AMX(K)-AMA(K)-AMB(K)-AMC(K)-AME(K)-AMF(K)-AMG(K)
      OX = OX - OXMAS
      GO TO 70
1001) IF (NC.EQ.4.AND.K.EQ.LOK) PRINT 8001,K
      GO TO 1200
1100) AIX(K) = AIX(K) + TEMPEN
      ETH = ETH + TEMPEN
      PDCSUM = PDCSUM + TEMPEN
      NIT = 8
      IF (NC.EQ.4.AND.K.EQ.LOK) PRINT 9000,NIT,AMX(K),AMA(K),AMB(K),
```

```
     1  AMC(K),AMD(K),AME(K),AMF(K),AMG(K),TEMPEN,AIX(K)
1200 CONTINUE
     PRINT 8002,PDCSUM
     RETURN
8000 FORMAT (1X,I2,10(1X,E12.5))
8001 FORMAT (1X,*NO COMBUSTION IN CELL *,I5)
8002 FORMAT (2X,*PDC ENERGY DEPOSITED = *,E16.8,/)
8003 FORMAT (1X,I5,9(1X,E12.5))
     END



     SUBROUTINE STEADY
C
C                    D  I  M  E  N  S  I  O  N
C
     DIMENSION AIX(6200),AMX(6200),P(6200),U(6200),V(6200),
    1AMA(6200),AMB(6200),AMC(6200),AMD(6200),AME(6200),AMF(6200),
    2AMG(6200),THETA(6200),DY(200),XX(101),YY(201)
     DIMENSION X(100),     Y(200),     Z(150),     IZ(150),    FLEFT(200)
     DIMENSION TAU(100),   PL(200),    UL(200),    PR(200),    UR(200)
     DIMENSION IW1(15),    YAMC(200),  SIGC(200),  GAMC(200),  DX(100)
     COMMON            Z       ,XX      ,UR      ,PR      ,YY
     COMMON            AIO     ,AIX     ,AM      ,THETA   ,AMX     ,AREA
     COMMON            BIG     ,BOUNCE  ,DDXN    ,DDYK    ,DKE     ,DYC
     COMMON            DX      ,DY      ,E       ,FD      ,FS      ,FX
     COMMON            OUT     ,P       ,PABOVE  ,PBLO    ,PIOTS   ,PPABOV
     COMMON            PRR     ,PUL     ,QDT     ,RC      ,REZ     ,RHO
     COMMON            RL,RR,SIG,QQQDFL,SWITCH ,TABLM
     COMMON            TAU     ,TAUDTS  ,TAUDTX  ,U       ,UK      ,URR
     COMMON            UT      ,UU      ,UUU     ,UTEF    ,UVMAX   ,V
     COMMON            VABOVE  ,VOLO    ,VEL     ,VC      ,VT      ,VTEF
     COMMON            VV      ,VVABOV  ,VVBLO   ,W2      ,W3      ,WPS
     COMMON            WS      ,WSA     ,WSB     ,WSC     ,XL      ,XLF
     COMMON            XN      ,XR      ,YL      ,YLW     ,YN      ,YU
     COMMON            ZMAX    ,I       ,II      ,IN      ,IR      ,IWS
     COMMON            IWSA    ,IWSB    ,IWSC    ,IW1     ,J       ,JN
     COMMON            JP      ,JR      ,K       ,KN      ,KP      ,KR
     COMMON            KR1     ,L       ,M       ,MA      ,MB      ,MC
     COMMON            MD      ,ME      ,MZ      ,N       ,NK      ,NKMAX
     COMMON            NK1     ,NO      ,NR      ,IW2
     COMMON   AMA,AMB,AMC,AMD,AME,AMF,AMG
C
C
C                    E  Q  U  I  V  A  L  E  N  C  E
C
     EQUIVALENCE       (Z,IZ,PROB),      (Z(2),CYCLE),     (Z(3),DT),
    1(Z(4),PRINTS),    (Z(5),PRINTL),    (Z(6),DUMPT),     (Z(7),CSTOP),
    2(Z(8),PID),       (Z(9),TMZ),       (Z(10),GAM),      (Z(11),GAMD),
    3(Z(12),GAMX),     (Z(13),FTM),      (Z(14),FFA),      (Z(15),FFB),
    4(Z(16),TMDZ),     (Z(17),TMXZ),     (Z(18),XMAX),     (Z(19),TXMAX),
    5(Z(20),TYMAX),    (Z(21),AMDM),     (Z(22),AMXM),     (Z(23),DMM),
    6(Z(24),DMIN),     (Z(25),FEF),      (Z(26),DTMA),     (Z(27),CVIS),
    7(Z(28),NPR),      (Z(29),NPRI),     (Z(30),NC),       (Z(31),NPC),
    8(Z(32),NRC),      (Z(33),IMAX),     (Z(34),IMAXA),    (Z(35),JMAX),
    9(Z(36),JMAXA),    (Z(37),KMAX),     (Z(38),KMAXA),    (Z(39),NMAX)
```

```
     0EQUIVALENCE      (Z(40),ND),       (Z(41),KDT),      (Z(42),IXMAX),
     1(Z(43),NOD),     (Z(44),NOPR),     (Z(45),NIMAX),    (Z(46),NJMAX),
     2(Z(47),I1),      (Z(48),I2),       (Z(49),I3),       (Z(50),I4),
     3(Z(51),N1),      (Z(52),N2),       (Z(53),N3),       (Z(54),N4),
     4(Z(55),N5),      (Z(56),N6),       (Z(57),N7),       (Z(58),N8),
     5(Z(59),N9),      (Z(60),N10),      (Z(61),N11),      (Z(62),NRM),
     6(Z(63),TRAD),    (Z(64),XNRG),     (Z(65),SN),       (Z(66),DXN),
     7(Z(67),RADER),   (Z(68),RADET),    (Z(69),RADEB),    (Z(70),DTRAD),
     8(Z(71),REZFCT),  (Z(72),RSTOP),    (Z(73),SHELL),    (Z(74),SBOUND),
     9(Z(75),TOZONE),  (Z(76),ECK),      (Z(77),SBOUND),   (Z(78),X1)
     0EQUIVALENCE      (Z(79),X2),       (Z(80),Y1),       (Z(81),Y2),
     1(Z(82),CABLN),   (Z(83),VISC),     (Z(84),T),        (Z(85),GMAX),
     2(Z(86),WSGO),    (Z(87),WSGX),     (Z(88),GMAJR),    (Z(89),GMAXR),
     3(Z(90),S1),      (Z(91),S2),       (Z(92),S3),       (Z(93),S4),
     4(Z(94),S5),      (Z(95),S6),       (Z(96),S7),       (Z(97),S8),
     5(Z(98),S9),      (Z(99),S10),
     7(Z(130),NFITR),  (Z(131),NFITT),   (Z(132),NFITR),
     8(Z(133),NPARB),  (Z(134),NPART),   (Z(135),NPARR)
C
      EQUIVALENCE(P(200),GAMC),(P(400),YAMC)
      EQUIVALENCE       (UR,UL,FLEFT),    (PR,PL,SIGC)
      EQUIVALENCE       (XX(2),X(1)),     (YY(2),Y(1))
C
      VEL = 0.0
      FLAG2 = 12.0
   20 FLAG1 = 0.0
   30 READ (4) PR(1), PR(2), N3
      WRITE (6,8006) PR(1),PR(2),N3
      IF (PR(1).EQ.555.0) 40,31
   31 IF (PR(1).EQ.666.0) 32,35
   32 WRITE (6,8009)
      GO TO 1000
   35 FLAG1 = FLAG1 + 1.0
      IF (FLAG1 - FLAG2) 34,34,9901
   34 WRITE (6,8100) FLAG1
      GO TO 30
   40 READ (4) (Z(I),I=1,3)
      WRITE (6,8104) (Z(I),I=1,3)
      READ (4) (DUM,DUM,DUM,DUM,P(I),DUM,I=1,KMAXA)
      WRITE (6,8106) (P(I),I=1,3)
      READ (4) DUM
      READ (4) DUM
      READ (4) DUM
      IF (VEL) 50,50,60
   50 DO 55 I = 1,KMAX
   55 AMX(I) = P(I)
      VEL = 1.0
      WRITE (6,8102)
      GO TO 30
   60 SUM = 0.0
      I = 3
   70 DO 75 J = 2,JMAX
      K = I + 1 + (J - 1)*IMAX
      DIFF = ABS(P(K)) - ABS(AMX(K))
   75 SUM = SUM + ABS(DIFF)
   80 WRITE (6,8002) PR(2), SUM
```

```
      IF (CSTOP - PR(2)) 1000,1000,90
   90 DO 95 I = 1,KMAX
   95 AMX(I) = P(I)
      GO TO 20
 9901 WRITE (6,8001) FLAG2
      GO TO 1000
 8001 FORMAT (22H0555 WAS NOT FOUND IN ,F5.1,7H CYCLES )
 8002 FORMAT (17H0CYCLE NUMBER IS ,F5.0,5X,22HSUM OF DIFFERENCES IS ,
     1  0PE12.6 //)
 8004 FORMAT (16H0666.0 WAS FOUND  /)
 8006 FORMAT(10X,16HFIRST RECORD IS ,2E12.4,I6,/)
 8100 FORMAT (10X,9HFLAG1 IS ,F5.1,/)
 8102 FORMAT (10X,14HVEL SET TO 1.0  /)
 8104 FORMAT (10X,25HTHE FIRST 3 Z VALUES ARE ,3E12.6,/)
 8106 FORMAT (10X,25HTHE FIRST 3 P VALUES ARE ,3E12.6,/)
 1000 REWIND 4
      RETURN
      END



      SUBROUTINE ERRER(NR,NK)
      CALL DISPLA(5H NR= ,NR)
      CALL TIME(14HERROR IN SHELL )
      CALL DISPLA(5H NK= ,NK)
      PRINT 8000, NR, NK
 8000 FORMAT (1H ,/,5X,*NR = *,I4,4X,*NK = *,I4,//)
      CALL PRINTZ
      STOP
      END
```

APPENDIX D

ADDITIONAL PLOTS FOR THE MASS AND ENERGY ADDITION PROBLEM

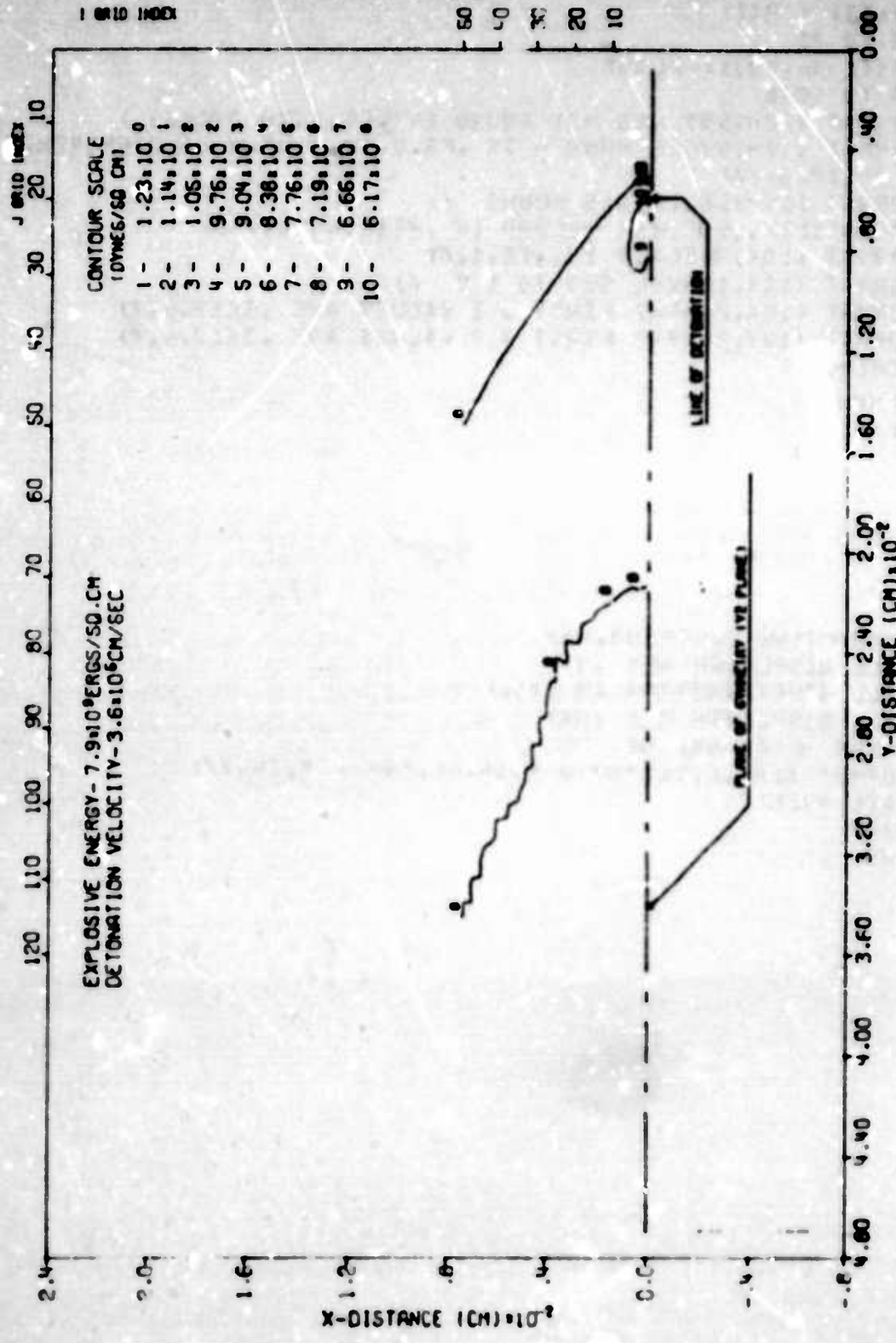


Figure D1. Constant Pressure Contours (Isobars)

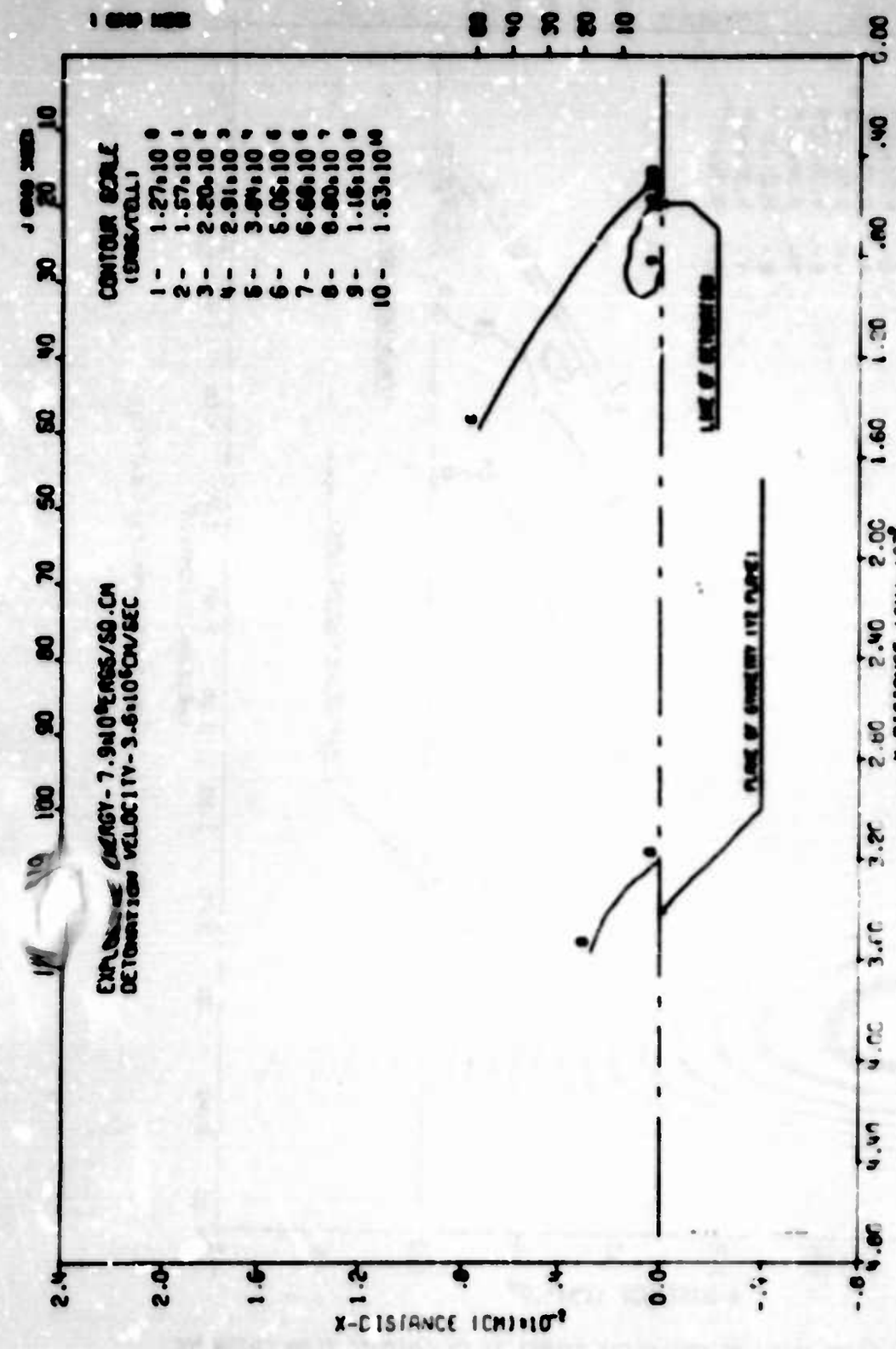


Figure D2. Constant Energy Contours

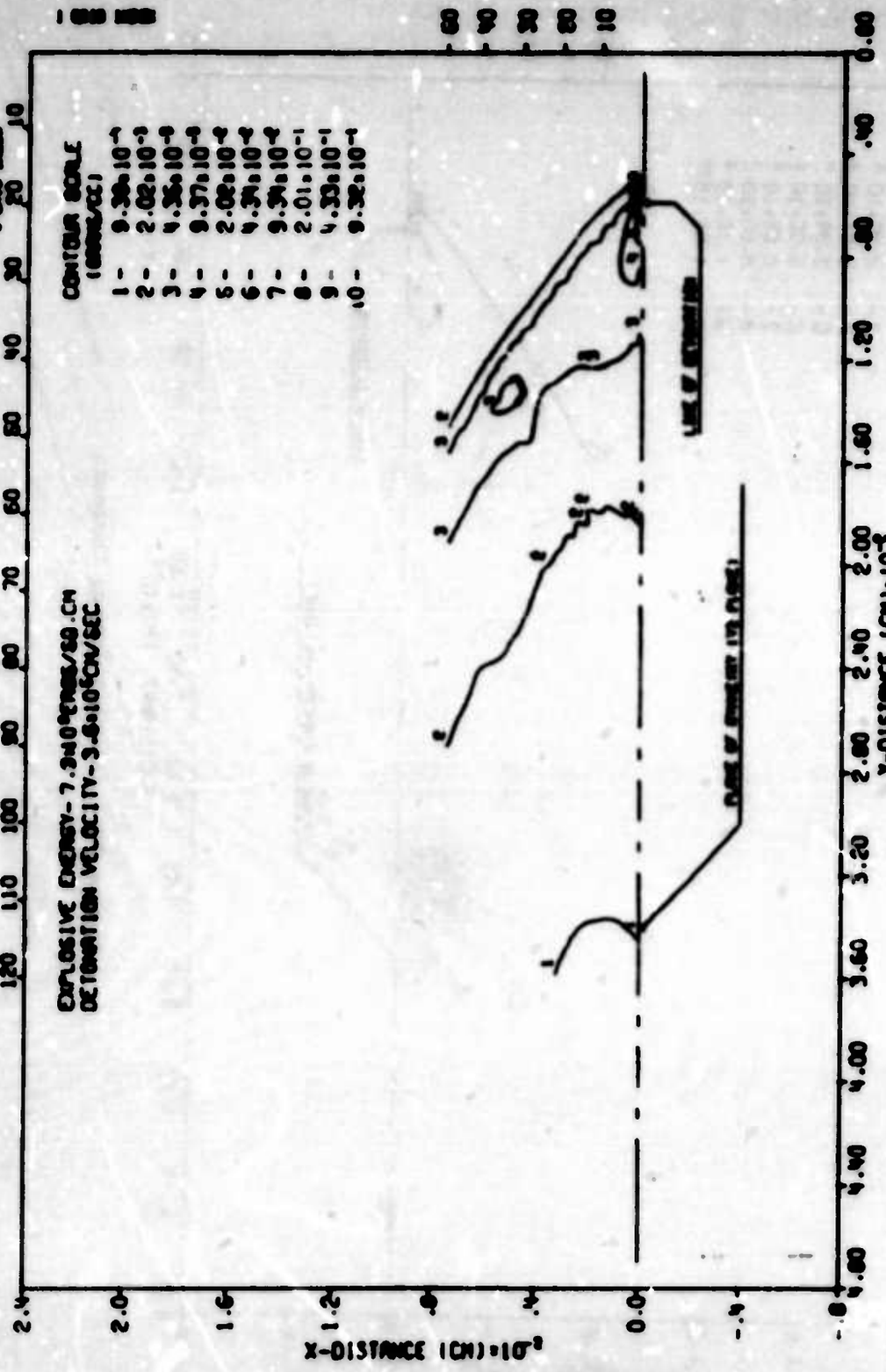


Figure D3. Constant Density Contours

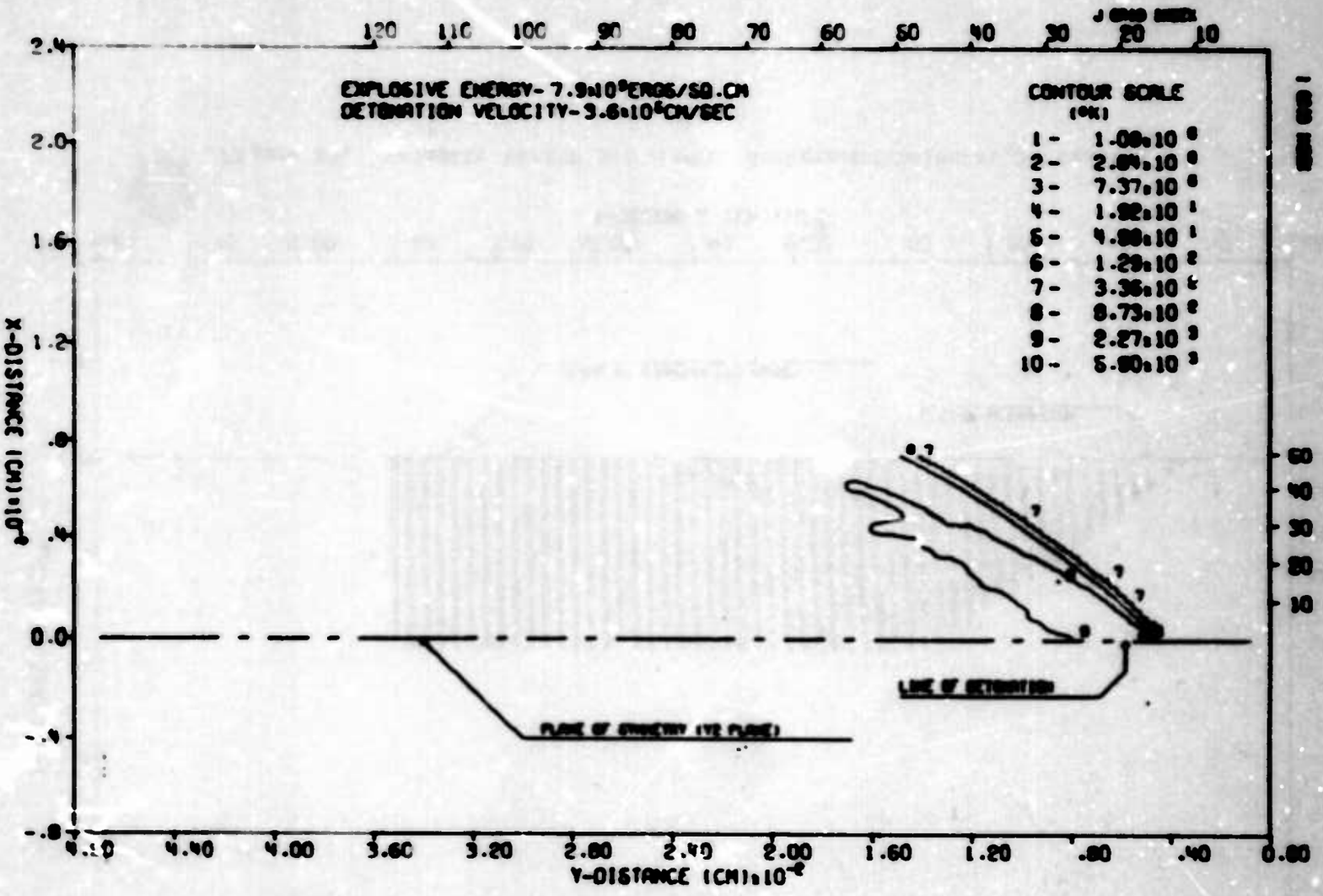
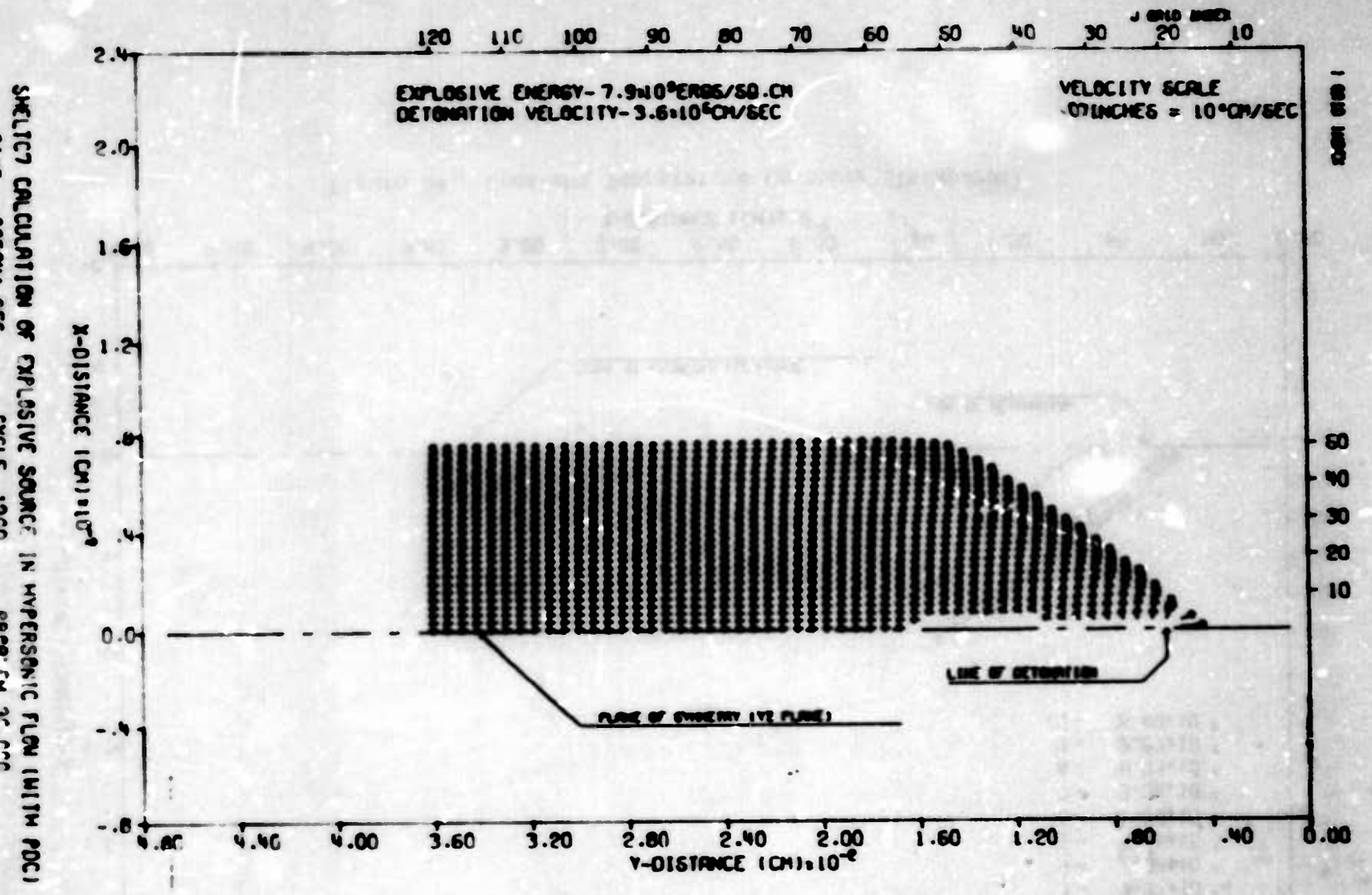


Figure D4. Constant Temperature Contours (Isotherms)



SHELICZ CALCULATION OF EXPLOSIVE SOURCE IN HYPERSONIC FLOW (WITH PDC)
TIME .0050011 SEC CYCLE 1200 PROBLEM 30.000

Figure D5. Velocity Vector Field with Background Velocity Subtracted

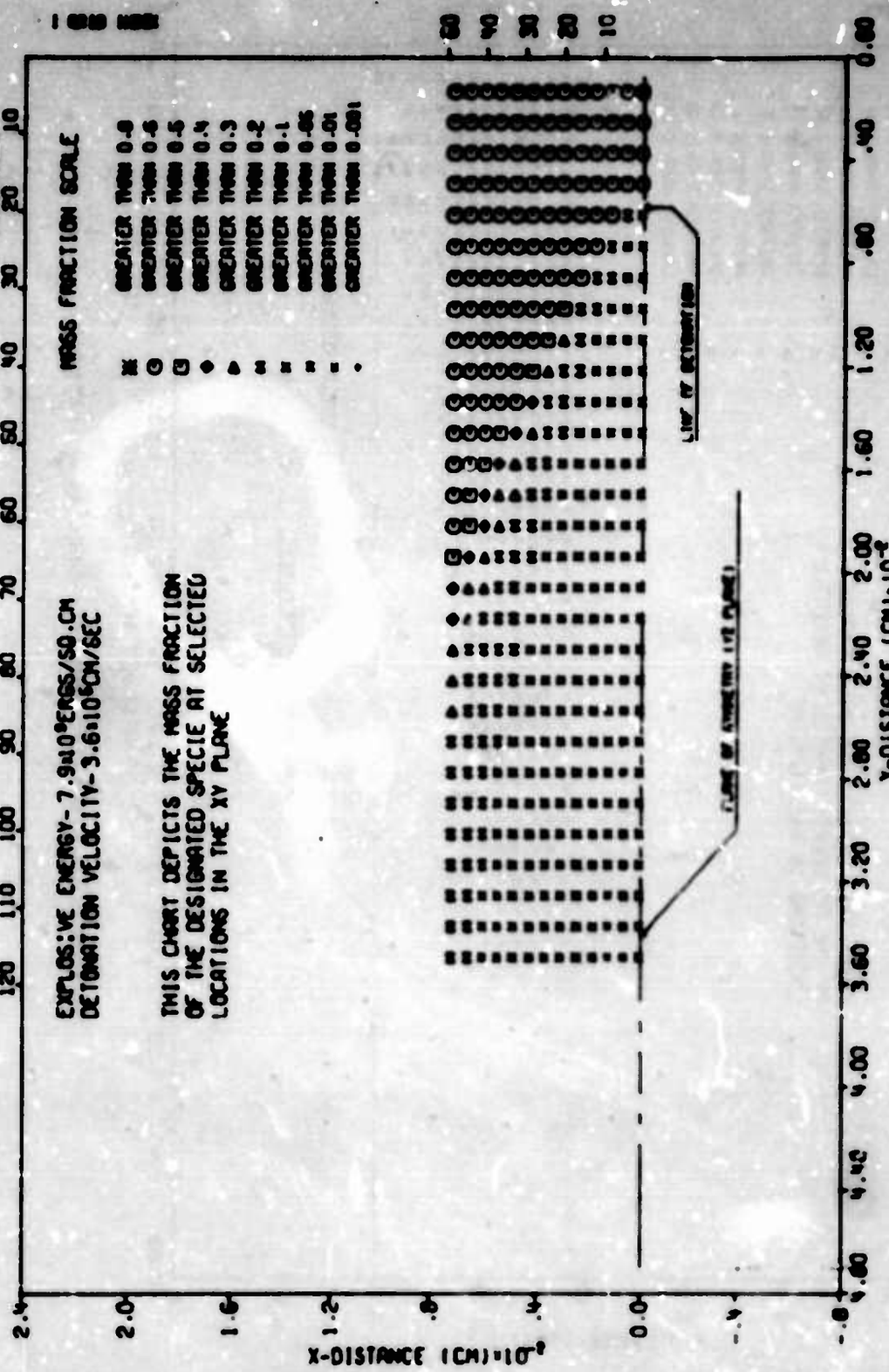


Figure 46. Mass Distribution--Species A (Nitrogen)

Figure D7. Mass Distribution--Species B (Oxygen)

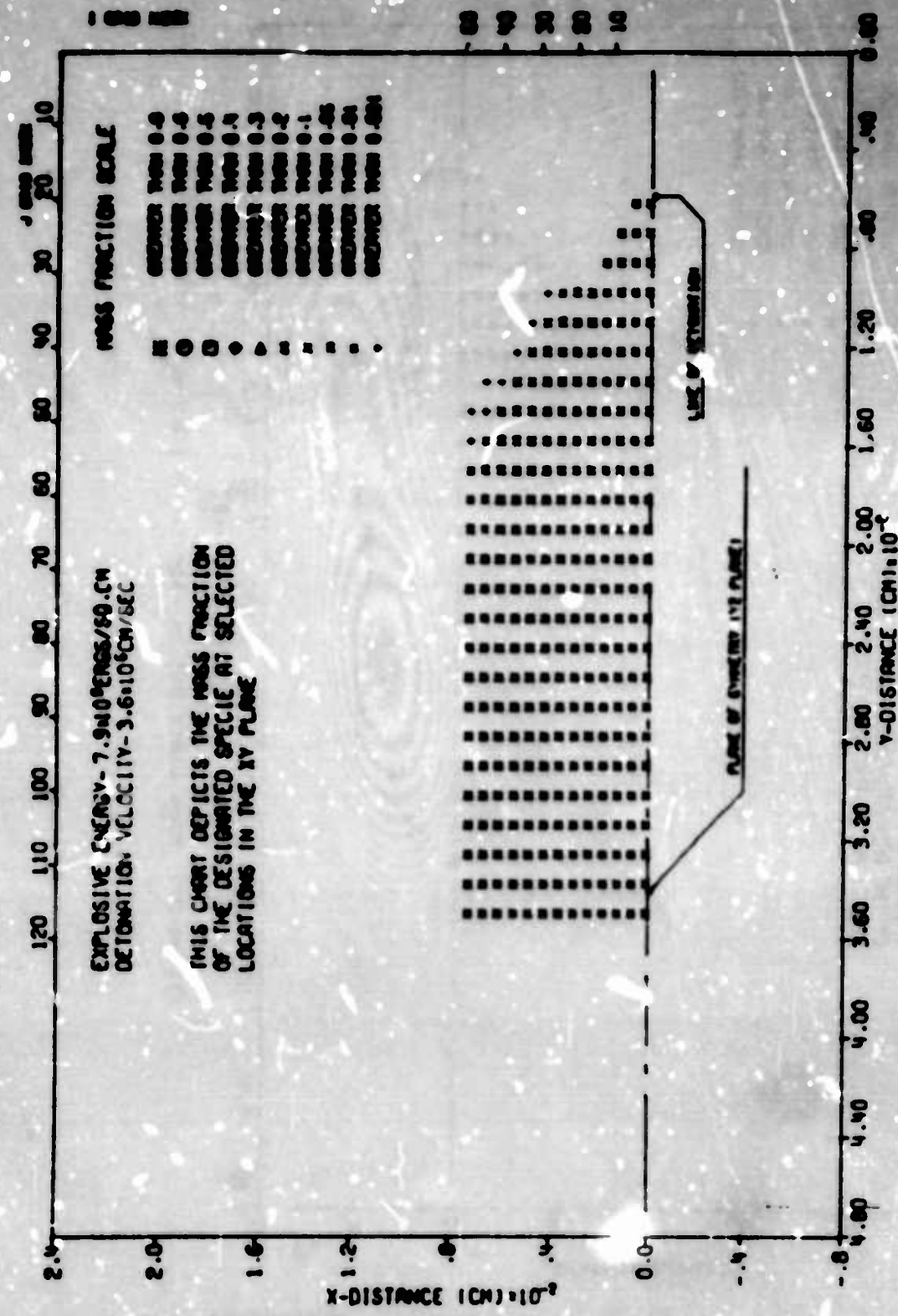


Figure D8. Mass Distribution--Species C (Carbon Monoxide)

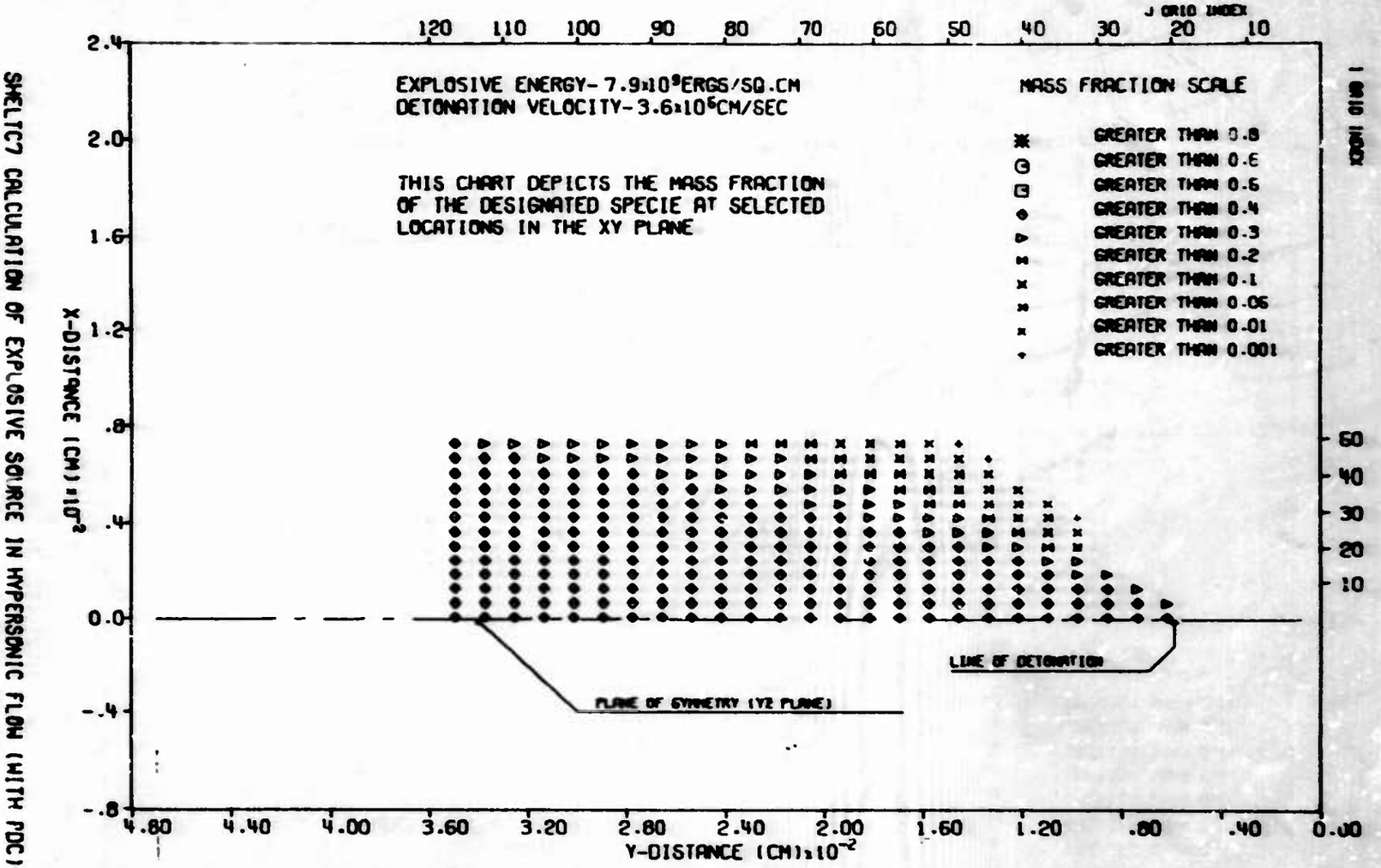


Figure D9. Mass Distribution--Species D (Carbon Dioxide)

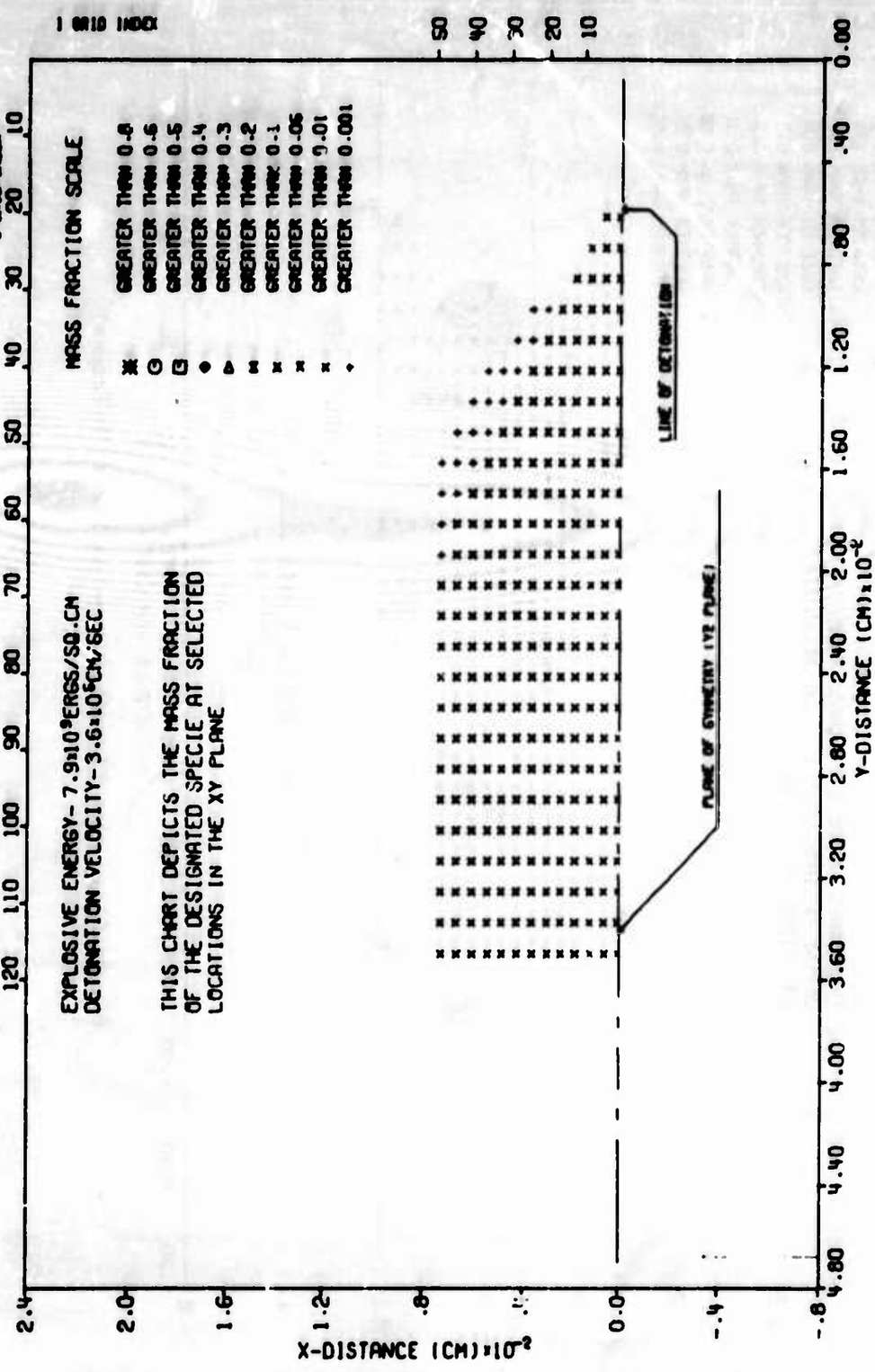


Figure D10. Mass Distribution--Species E (Water)

SHELTC7 CALCULATION OF EXPLOSIVE SOURCE IN HYPERSONIC FLOW (WITH PDC)
TIME .005041 SEC CYCLE 1200 PROBLEM 30.000

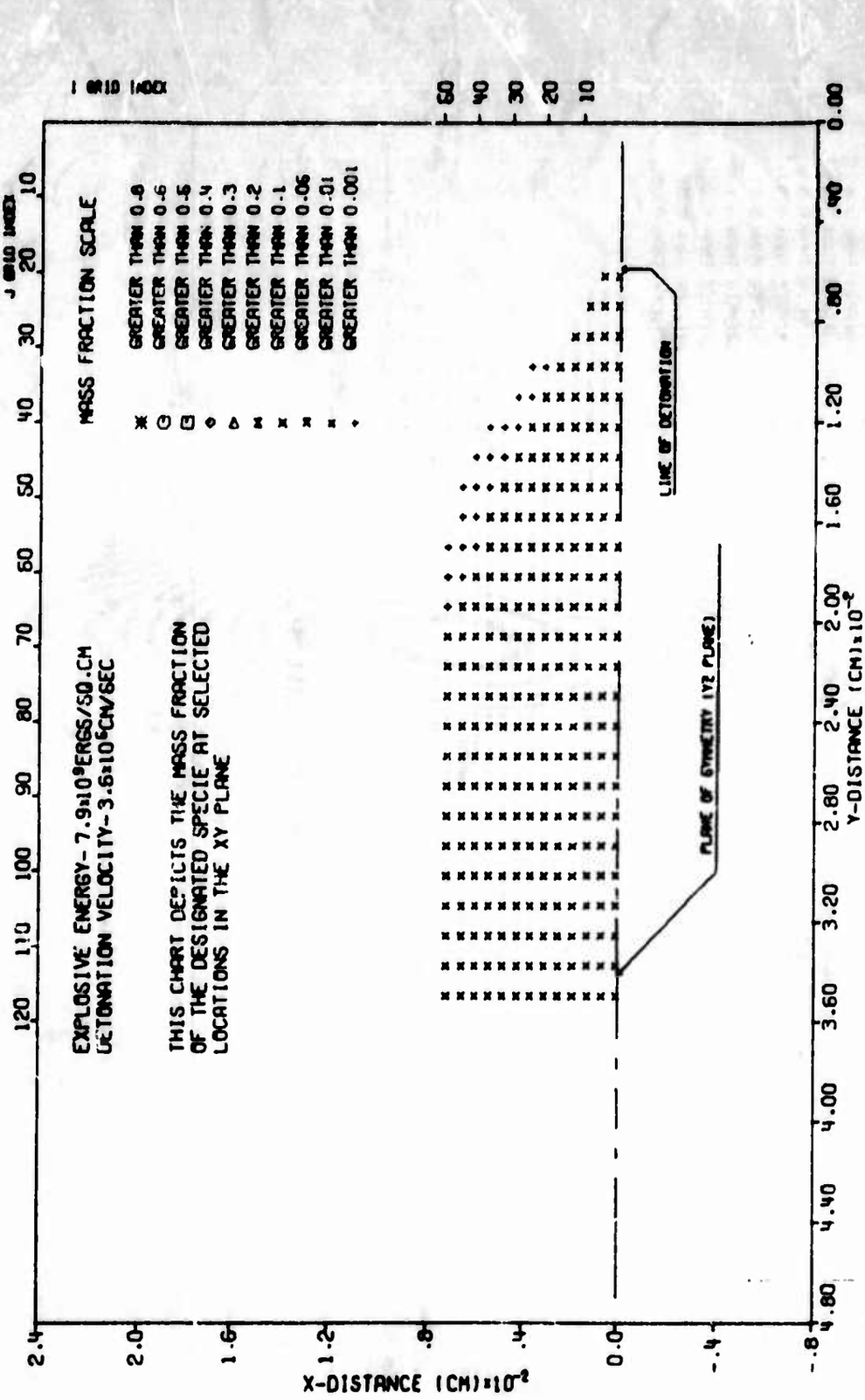


Figure D11. Mass Distribution--Species F (Hydrogen)

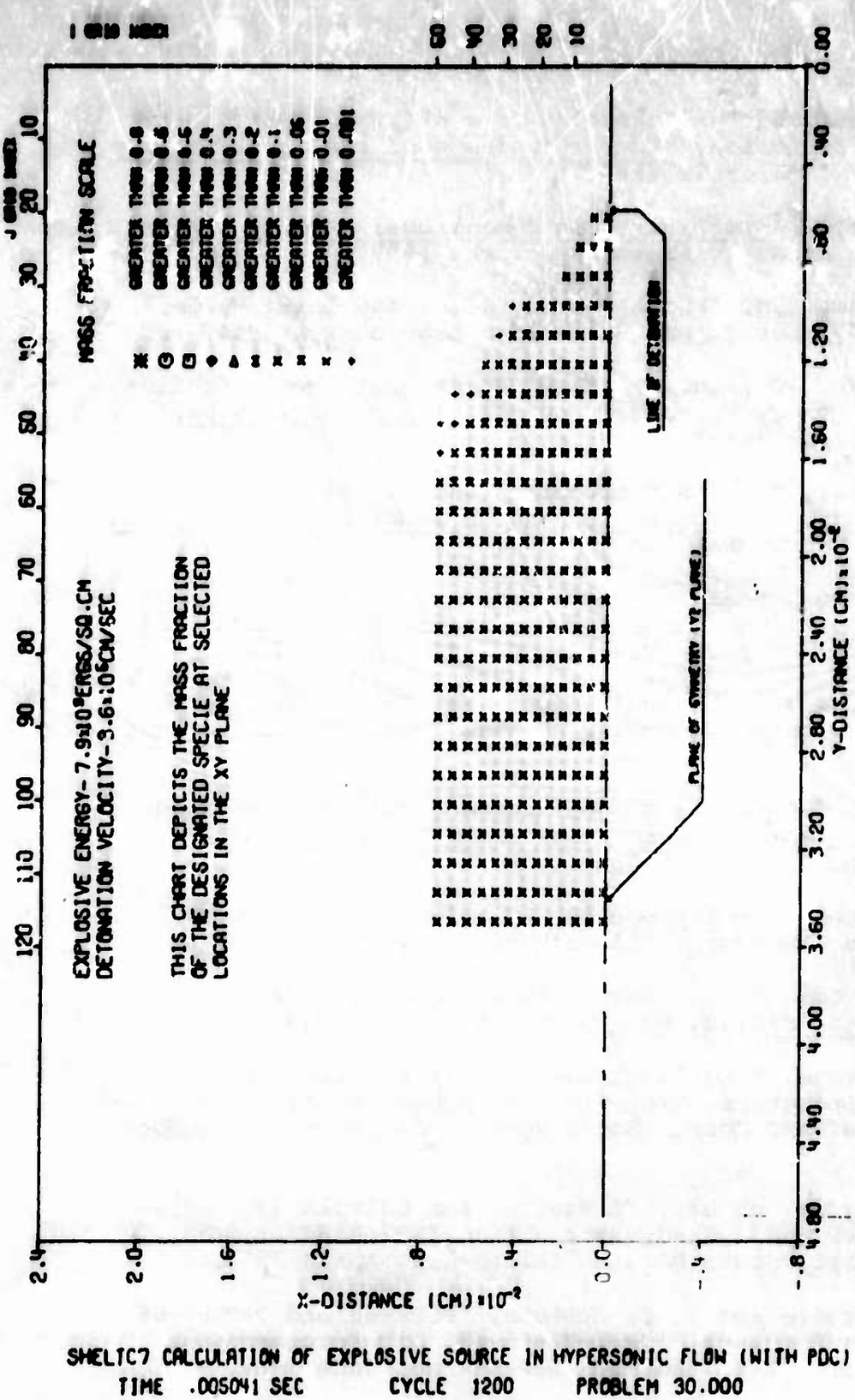


Figure D12. Mass Distribution--Species G (Carbon)

SHELTC7 CALCULATION OF EXPLOSIVE SOURCE IN HYPERSONIC FLOW (WITH PDC)
TIME .005041 SEC CYCLE 1200 PROBLEM 30.000

REFERENCES

1. G. I. Taylor, "The Formation of a Blast Wave by a Very Intense Explosion," The Proceedings of the Royal Society, Vol. 201A, March 22, 1950.

2. L. I. Sedov, Similarity and Dimensional Methods in Mechanics, Academic Press, New York, 1959.

3. J. Von Neumann, "Shock Hydrodynamics and Blast Waves," AECD-2860, Los Alamos Scientific Laboratory, 1944.

4. S. C. Lin, "Cylindrical Shock Waves Produced by Instantaneous Energy Release," Journal of Applied Physics, Vol. 25, pp. 54-57, 1954.

5. A. Sakurai, "On the Propagation and Structure of the Blast Wave, I," Journal of the Physical Society of Japan, Vol. 8, No. 5, Sep.-Oct., 1953.

6. J. Von Neumann and R. D. Richtmeyer, "A Method for the Numerical Calculation of Hydrodynamic Shocks," Journal of Applied Physics, Vol. 21, p. 232, March 1950.

7. H. L. Brode, "Numerical Solution of Spherical Blast Wave," Journal of Applied Physics, Vol. 26, No. 6, June 1955.

8. H. H. Goldstine and J. Von Neumann, "Blast Wave Calculations," Communications on Pure and Applied Math, Vol. VIII, pp. 327-354, 1955.

9. H. L. Brode, "Point Source Explosion in Air," RM-1824-AEC, The RAND Corp., Santa Monica, California, 1956.

10. H. L. Brode, "Blast Wave from a Spherical Charge," The Physics of Fluids, Vol. 2, No. 2, March 1959.

11. H. L. Brode, "The Blast Wave in Air Resulting from a High Temperature, High Pressure Sphere of Air," RM-1825-AEC, The RAND Corp., Santa Monica, California, December 1956.

12. H. L. Brode, et al., "A Program for Calculating Radiation Flow and Hydrodynamic Motion," RM-5187-PR, The RAND Corp., Santa Monica, California, April 1967.

13. R. N. Brodie and J. E. Hormuth, "PUFF-66 and P-PUFF-66 Computer Programs," AFWL-TR-66-48, Air Force Weapons Laboratory, Kirtland AFB, New Mexico, June 1966.

14. M. Lutzky, "The Flow Field Behind a Spherical Detonation in TNT Using the Landau-Stanyukovich Equation of State for Detonation Products," NOLTR 64-40, U.S. Naval Ordnance Lab., White Oak, Md., Feb. 1965.

15. F. H. Harlow, "The Particle-in-Cell Computing Method for Fluid Dynamics," Methods in Computational Physics, Vol. 3, p. 319, Academic Press, New York, 1964.

16. W. A. Whittacker, et al., "Theoretical Calculations of the Phenomenology of HE Detonations," AFWL TR-66-141, Vol. I, Air Force Weapons Lab., Kirtland AFB, New Mexico, November 1966.

17. W. E. Johnson, "OIL-A Continuous Two-Dimensional Hydrodynamic Code," GAMD-5580, General Atomic/General Dynamics, San Diego, California, October 1964.

18. E. A. Nawrocki, et al., "Theoretical Calculations of the Phenomenology of the Distant Plain Event 6," AFWL TR-67-57, Air Force Weapons Lab., Kirtland AFB, New Mexico, 1967.

19. C. E. Needham, et al., "Theoretical Calculations of the Detonation of a 1000 Pound Sphere of TNT at 15 Feet Above Ground Level," AFWL TR-66-128, Air Force Weapons Lab., Kirtland AFB, New Mexico, 1966.

20. C. E. Needham and T. Burghard, "Airblast Calculation, Event MINEUNDER," AFWL TR-69-105, Air Force Weapons Lab., Kirtland A B, New Mexico, 1969.

21. J. G. Trulio, "Theory and Structure of the AFTON Codes," AFWL TR-66-19, Air Force Weapons Lab., Kirtland AFB, New Mexico, June 1966.

22. C. N. Kingery, "Air Blast Parameters Versus Distance for Hemispherical TNT Surface Bursts," BRL Report No. 1344, Ballistics Research Lab., Aberdeen, Md., 1966.

23. J. W. Reed, "Operation SAILOR HAT-Long Range Air Blast," DASA POR 4057, Defense Atomic Support Agency, Washington, D.C., 1966.

24. H. E. Auld, G. P. D'Arcy, and G. G. Leigh, "Simulation of Air-Blast-Induced Ground Motions," WLRT-65-26, Vol. I, Air Force Weapons Lab., Kirtland AFB, New Mexico, April 1965.

25. W. S. Filler, "Explosions in Enclosed Spaces," NAVORD Report 3890, U.S. Naval Ordnance Lab., White Oak, Md., Jan. 1956.

26. V. P. Korobeynikov, et al., "The Theory of Point Explosions," JPRS:14,334-U.S. Dept. of Commerce, Office of Technical Services, Joint Publications Research Services, Washington, D.C., 1962.

27. R. J. Swigart, "Third-Order Blast Wave Theory and its Application to Hypersonic Flow Past Blunt-Nosed Cylinders," Lockheed Aircraft Corp., Space and Missile Division, Palo Alto, Calif., July 1960.

28. E. F. Brocher, "On Similar Solutions for Strong Blast Waves and Their Application to Steady Hypersonic Flow," Journal of Aerospace Sciences, June 1962.

29. W. D. Hayes and R. F. Probstein, Hypersonic Flow Theory, Vol. 1, Academic Press, New York, 1966.

30. J. Von Mises, Mathematical Theory of Compressible Flow, Academic Press, New York, 1958.

31. P. I. Chushkin and L. Li-Kan, "Determination of the Parameters of Supersonic Two-Dimensional Flows of Gas," translation of a Russian paper, FTD-HT-66-465/1+2+4, Foreign Technology Division, Wright-Patterson AFB, Ohio, Dec. 1963.

32. G. G. Leigh and J. Yagla, "A Study of Shock Structure from Hydrodynamic Codes," to be published.

33. L. Lees and T. Kubota, "Inviscid Hypersonic Flow Over Blunt-Nosed Slender Bodies," Journal of Aerospace Science, Vol. 24, pp. 195-202, 1957.

34. M. A. Cook, The Science of High Explosives, Reinhold Publishing Corp., New York, 1958.

35. R. Durrett, et al., "The INSHALLA Calculation," AFWL TR-69-85, Air Force Weapons Laboratory, Kirtland AFB, New Mexico, 1969.

36. O. A. Hougen and K. M. Watson, Chemical Process Principles, Vol. I, John Wiley and Sons, New York, 1943.

37. R. A. Strehlow, Fundamentals of Combustion, International Textbook Co., Scranton, Penn., 1968.

38. JANAF Thermochemical Tables, Clearing House for Federal Scientific and Technical Information, U.S. Dept. of Commerce, Springfield, Virginia.

39. J. A. Owczarek, Fundamentals of Gas Dynamics, International Textbook Co., Scranton, Penn., 1964.

40. R. B[illegible]rk, N. Brooks, and R. Papetti, "A Numerical Technique for Solution of Multidimensional Hydrodynamic Problems," RM-2628-PR, The RAND Corp., Santa Monica, California, December 1963.

41. G. A. Hawkins and J. M. Smith, "Equilibrium Constants for Seventeen Gas Reactions," Research Series No. 108, The Engineering Experiment Station, Purdue University, Lafayette, Indiana, May 1949.

42. S. S. Cherry, et al., "Identification of Important Chemical Reactions in Liquid Propellant Rocket Engines," Pyrodynamics, Vol. 6, pp. 275-296, Gordon & Breach, Science Publishers Ltd., Northern Ireland, 1968.

43. J. M. Dewey, "The Air Velocity in Blast Waves from TNT Explosions," Proceedings of the Royal Society, A, Vol. 279, pp. 366-385, 1964.

44. T. A. McArtor, "Aerodynamic Interaction Effects Near a 2-D Sonic Jet Exhausting Normal to a Flat Plate Into a Mach 2.96 Free Stream," Masters Thesis, Arizona State U., Tempe, Ariz., Feb. 1971.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Air Force Weapons Laboratory (DEV)
Kirtland Air Force Base New Mexico, 87117

2a. REPORT SECURITY CLASSIFICATION

UNCLASSIFIED

2b. GROUP

3. REPORT TITLE

A CALCULATION OF THE BLAST WAVE FROM THE CONSTANT VELOCITY DETONATION OF AN EXPLOSIVE SHEET

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

July 1969-June 1971

5. AUTHOR(S) (First name, middle initial, last name)

Gerald G. Leigh, Maj, USAF

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO OF REFS
November 1971	276	44

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO. 06CB	AFWL-TR-71-137
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

Distribution limited to US Government agencies only because test and evaluation information is discussed in the report (Nov 71). Other requests for this document must be referred to AFWL (DEV), Kirtland AFB, NM, 87117.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	AFWL (DEV) Kirtland AFB, NM 87117

13. ABSTRACT

(Distribution Limitation Statement B)

Predictions of the blast wave from the constant velocity detonation of a thin explosive sheet have not previously considered the contribution of the postdetonation combustion of the detonation products. This problem is simplified by first considering the explosive material as releasing only energy into an ideal gas medium. This is solved for several energy densities and velocities of detonation. An expression that can predict the shape and location of the shock front from a line source of energy in a hypersonic flow is then developed. The problem is more realistically modelled by considering the explosive sheet to release six distinct mass species, as well as energy, into a mixture of nitrogen and oxygen. Results of this study shift the shock front to a position inside the previously determined shock front. Using the dissociation of molecular oxygen as the determinant for the reaction rate of combustion, and models for material mixing, the contribution of postdetonation combustion (PDC) to the total energy is determined. The mass and energy addition problems were then recomputed with PDC included. Results indicate that while significant amounts of energy are released by PDC, they are small compared to the energy released by the detonation, and there is essentially no contribution from PDC in the early time formation of the blast wave.

DD FORM 1 NOV 68 1473

UNCLASSIFIED
Security Classification

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Civil engineering						
Blast wave						
Explosive sheet						
Postdetonation combustion						

UNCLASSIFIED

Security Classification